





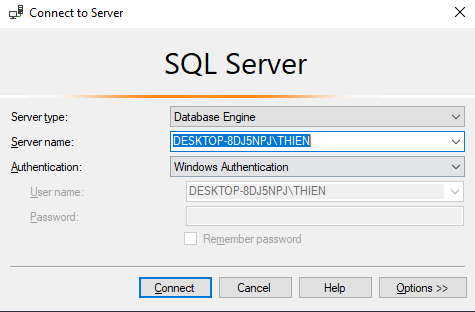
**Sinh viên thực hiện : Trần Kim Thiện**

**MSSV : 1050080076**

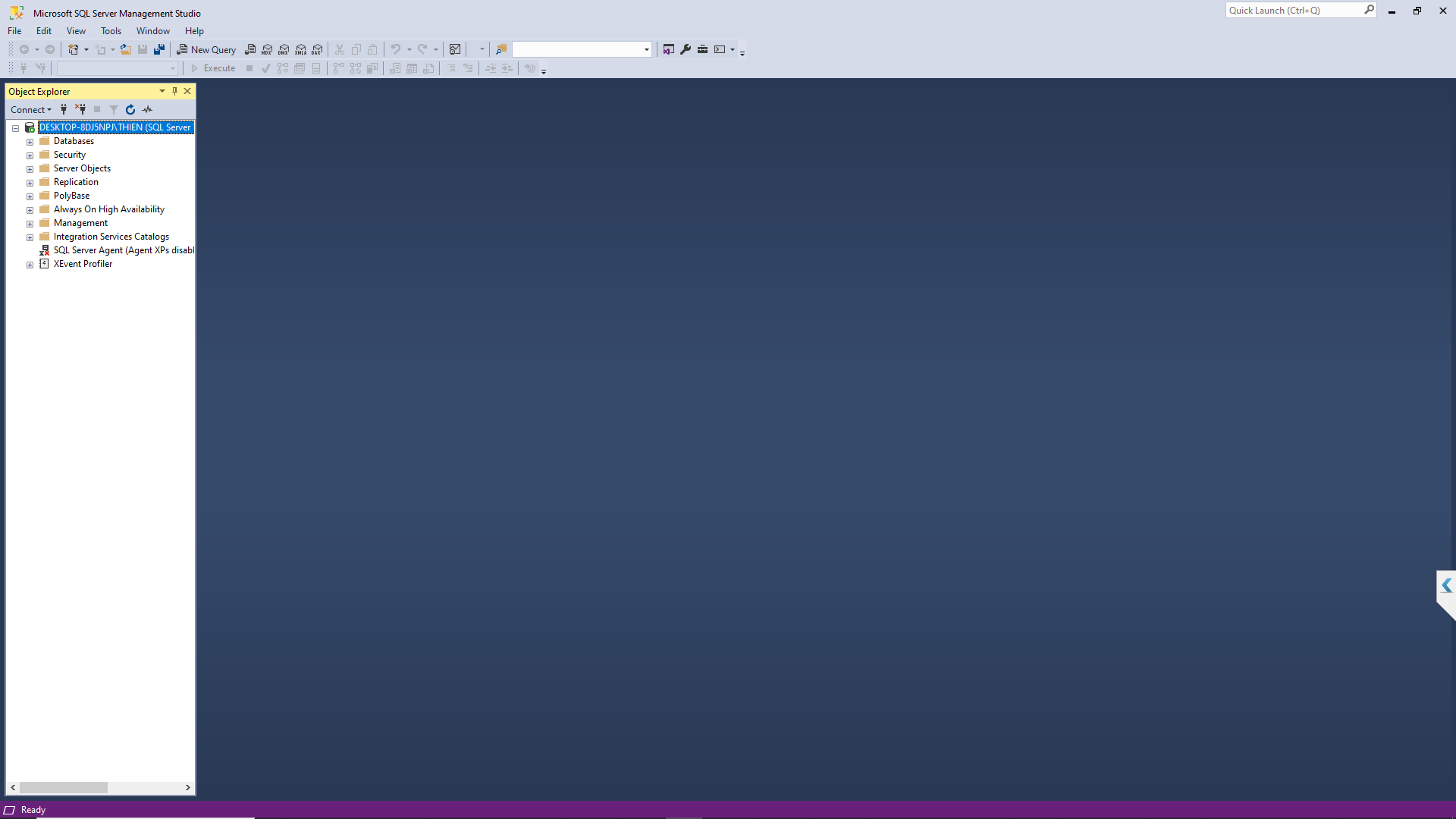
**Lớp : 10\_ĐH\_CNTT2**

Module 1: **Phần 1: SQL Server Management Studio**

Start🡪 Program File🡪 MicroSoft SQL Server 2008🡪SQL Server Management  
studio

****

Màn hình khi đã Connect vào :

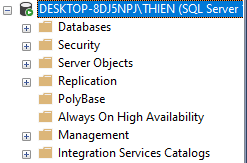


▪ Có bao nhiêu SQL Server Group, mỗi Server tên là gì? Đang connect hay  
disconnect?  
Có 1 SQL Server Group : Tên là DESKTOP-8DJ5NPJ



SQL Server - Đang connect

▪ Liêt kê các thành phần trong Server hiện hành



Gồm :

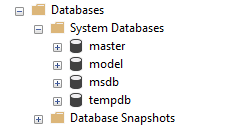
* Databases
* Security
* Server Objects
* Replication
* PolyBase
* Always On High Availability
* Management
* Integration Services Catalogs

▪ Trong server hiện hành, có các Database nào?

Trong server hiện hành, có 2 Database :

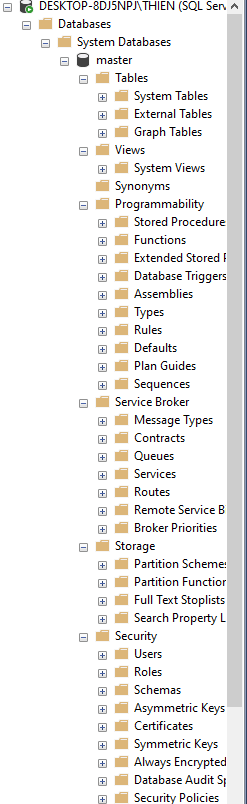


System Databases và Database Snapshots  
   
▪ Trong mỗi Database có những đối tượng nào?



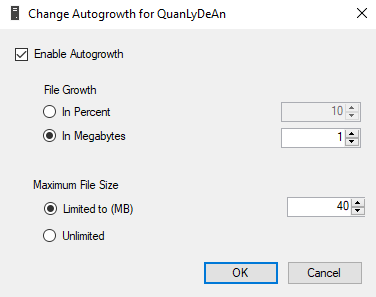
System Databases có Master , model , msdb , tempdb và Database Snapshots thì trống rỗng chưa có dữ liệu

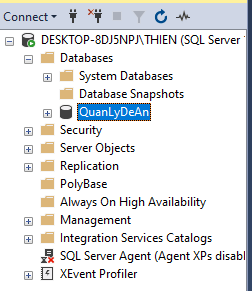
Khảo sát : Tables và các thành phần còn lại



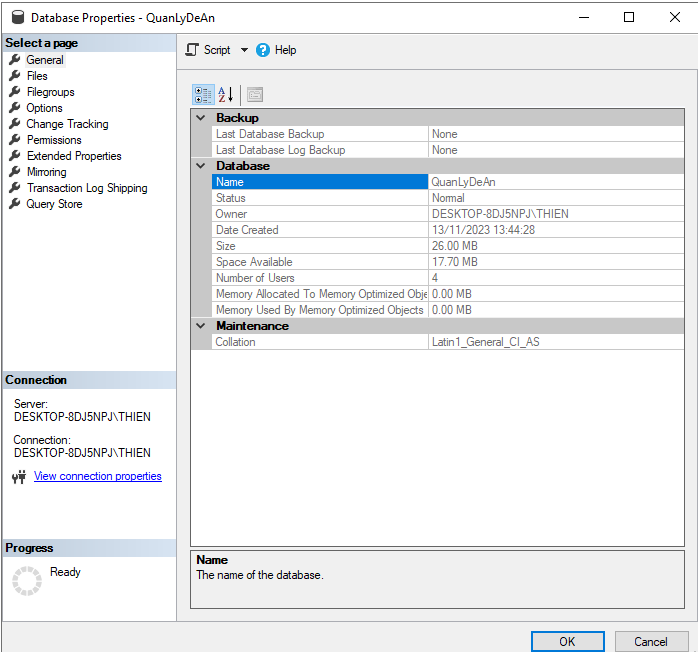
**Phần 2: Tạo và quản lý CSDL**

Tạo database tên QuanLyDeAn và giới hạn kích thước của chúng :

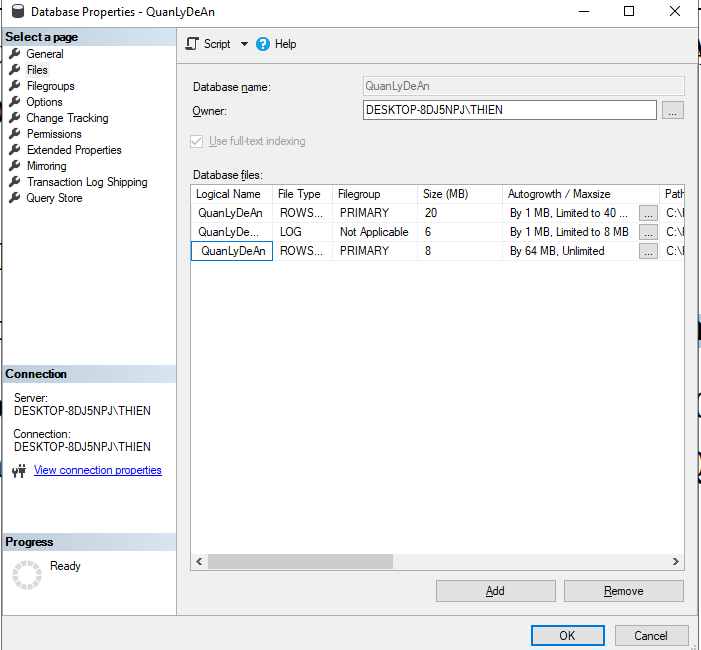




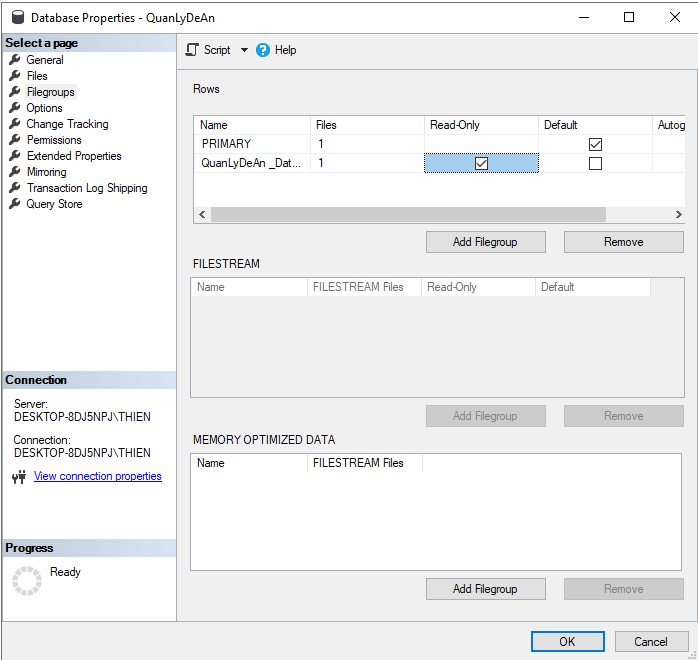
Xem lại thuộc tính (properties) của CSDL QuanLyDeAn. Thấy có Backup : last Database Backup , Log Backup , Tên , Status ,Owner , Date Created , Size , Space Avvailable …



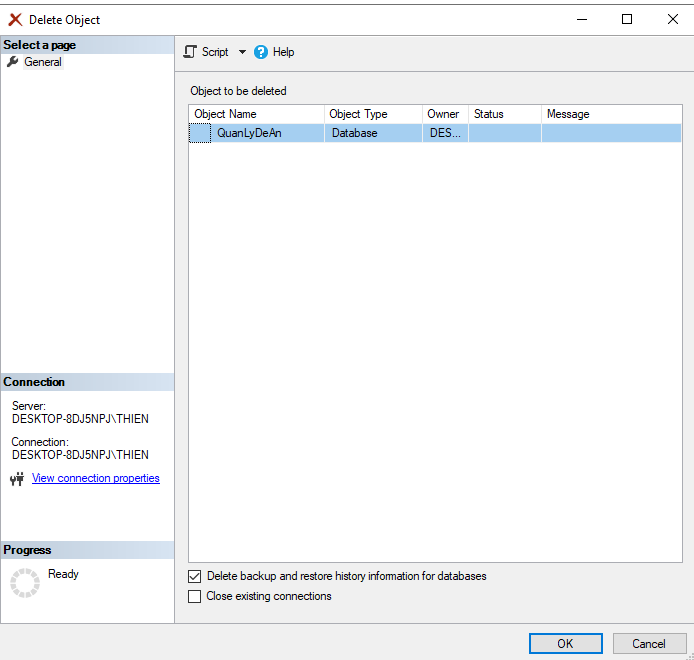
Tại cửa sổ properties của CSDL, khai báo thêm  
▪ Một Group File mới có tên là DuLieu QuanLyDeAn  
▪ Một tập tin dữ liệu (data file) thứ hai nằm trong Group file vừa tạo ở trên  
và có thông số như sau Tên login của data file là QuanLyDeAn \_Data2; Tên tập tin và đường dẫn vật lý của data file là T:\HoTenSV\ QuanLyDeAn  
\_Data2.ndf.  
▪ Chọn thuộc tính ReadOnly, sau đó đóng cửa sổ properies. Quan sát màu  
sắc của CSDL. Bỏ thuộc tính ReadOnly.



Thay đổi QuanLyDeAn \_Data2 trong Filegroup thành Read-Only



Xóa QuanLyDeAn



Dùng lệnh Create DataBase, tạo một CSDL với các tham số được liệt kê như  
trong bảng dưới. Lưu ý rằng CSDL này gồm một data file và nó được nằm  
trong primary filegroup

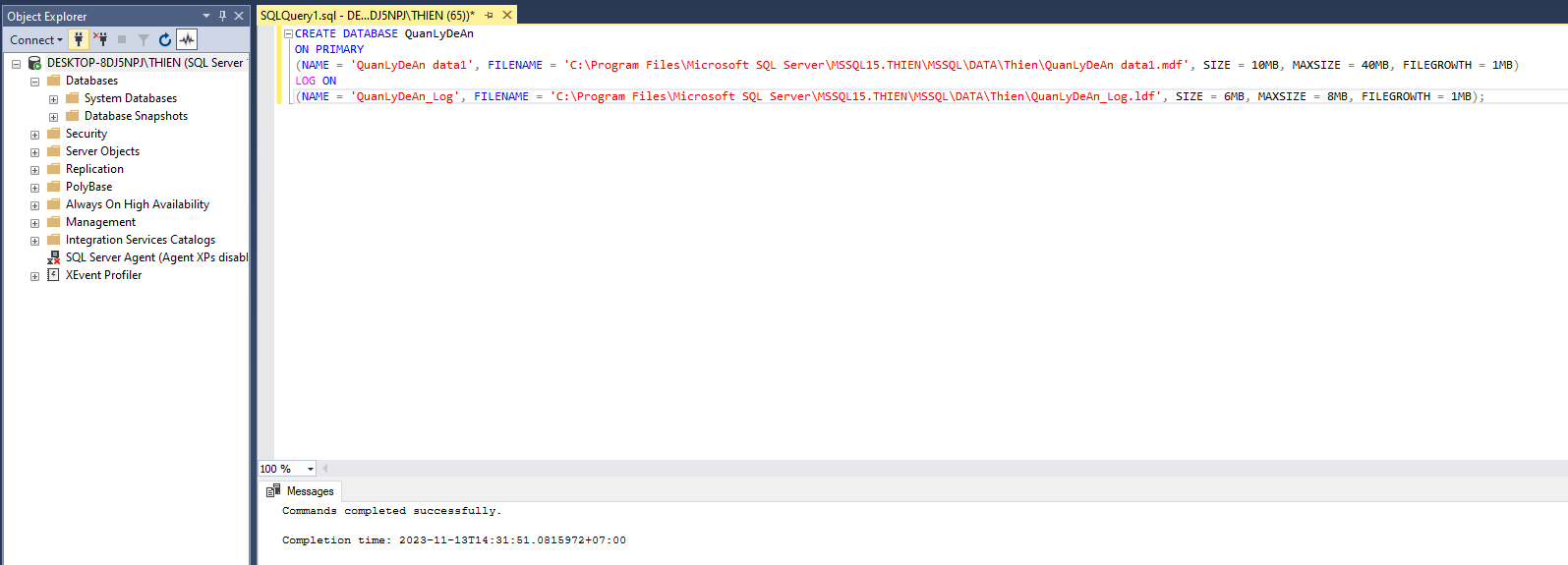
CREATE DATABASE QuanLyDeAn

ON PRIMARY

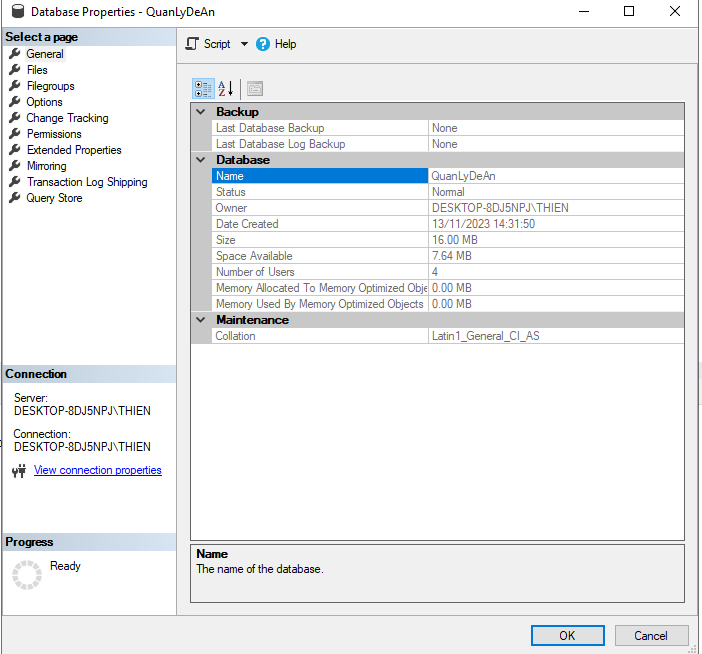
(NAME = 'QuanLyDeAn data1', FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.THIEN\MSSQL\DATA\Thien\QuanLyDeAn data1.mdf', SIZE = 10MB, MAXSIZE = 40MB, FILEGROWTH = 1MB)

LOG ON

(NAME = 'QuanLyDeAn\_Log', FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.THIEN\MSSQL\DATA\Thien\QuanLyDeAn\_Log.ldf', SIZE = 6MB, MAXSIZE = 8MB, FILEGROWTH = 1MB);



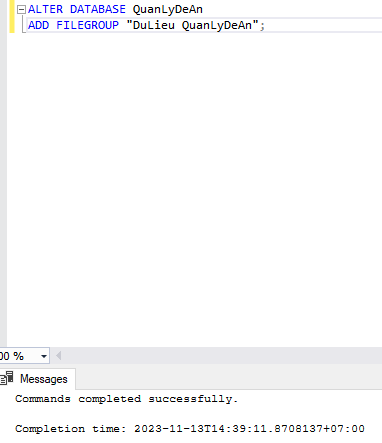
Xem lại thuộc tính của CSDL QuanLyDeAn bằng Management Studio và bằng  
thủ tục hệ thống sp\_helpDb, sp\_spaceused.



Thêm một filegroup có tên là DuLieu QuanLyDeAn (HD: dùng lệnh Alter  
DataBase <Tên Database> ADD FILEGROUP <Tên filegroup>)

ALTER DATABASE QuanLyDeAn

ADD FILEGROUP "DuLieu QuanLyDeAn";

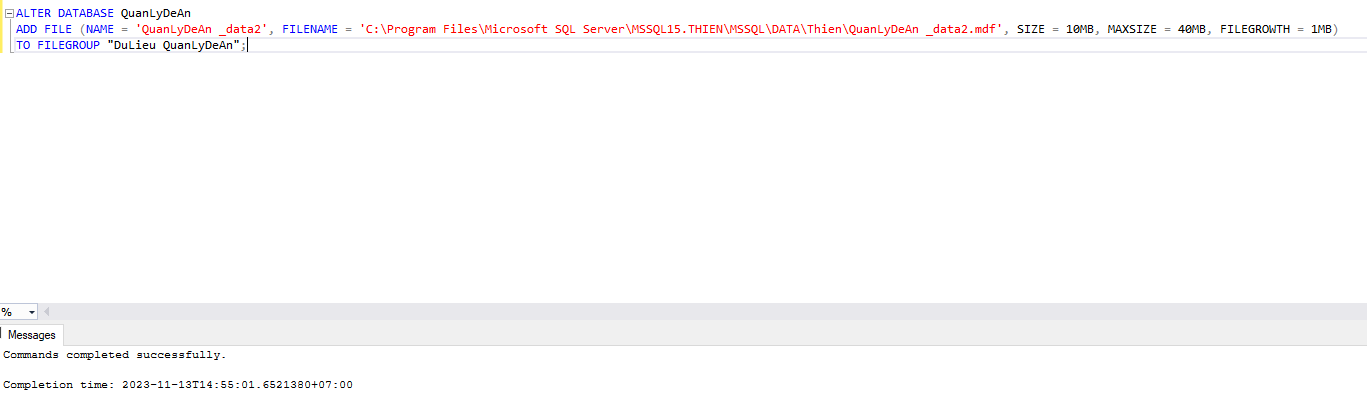


Khai báo một secondary file có tên logic là QuanLyDeAn \_data2, tên vật lý  
QuanLyDeAn \_data2.ndf nằm ở T:\HoTenSV, các thông số khác tùy bạn chọn,  
data file này nằm trong file group là QuanLyDeAn. (HD: Dùng lệnh Alter  
Database …. ADD FILE …. TO FILEGROUP …)

ALTER DATABASE QuanLyDeAn

ADD FILE (NAME = 'QuanLyDeAn \_data2', FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.THIEN\MSSQL\DATA\Thien\QuanLyDeAn \_data2.mdf', SIZE = 10MB, MAXSIZE = 40MB, FILEGROWTH = 1MB)

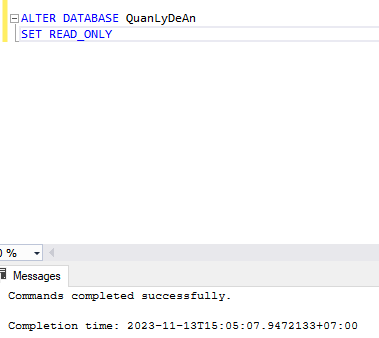
TO FILEGROUP "DuLieu QuanLyDeAn";

****

**e. Dùng Books Online, bạn tìm hiểu thủ tục hệ thống sp\_helpfilegroup dùng để làm gì?**

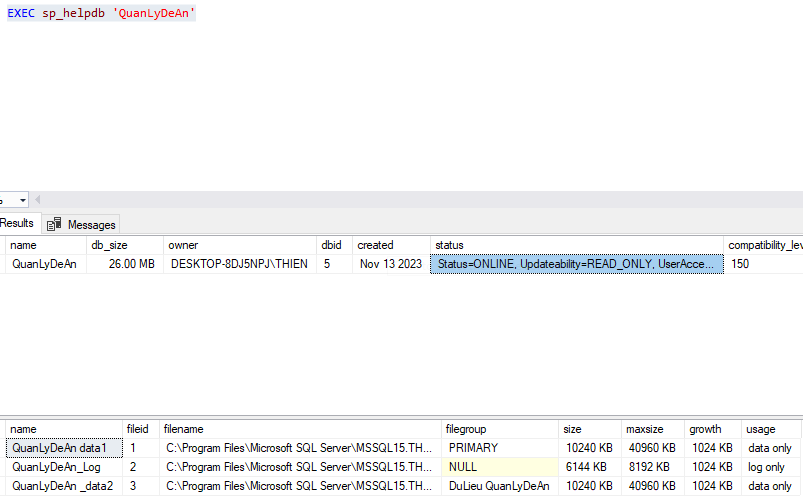
sp\_helpfilegroup là một stored procedure hệ thống trong SQL Server, trả về tên và thuộc tính của các filegroup liên quan đến cơ sở dữ liệu hiện tại. Nó được sử dụng để xác định tên của các filegroup để gắn kết hoặc tách khỏi máy chủ.

**f. Dùng lệnh Alter Database … Set … để cấu hình cho CSDL QuanLyDeAn có  
thuộc tính là Read\_Only. Dùng sp\_helpDB để xem lại thuộc tính của CSDL.  
Hủy bỏ thuộc tính Read\_Only.**

****

EXEC sp\_helpdb 'QuanLyDeAn'

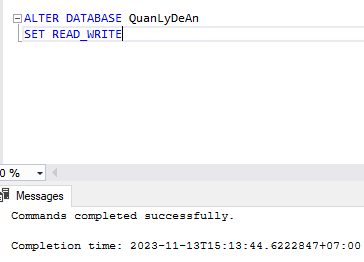
**Để hiện thị kiểm tra**

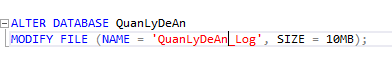
****

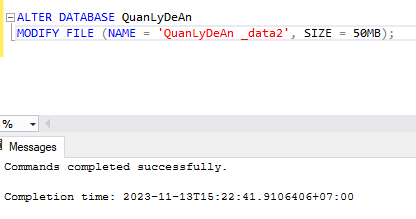
**Dùng để set lại Read\_Write :**

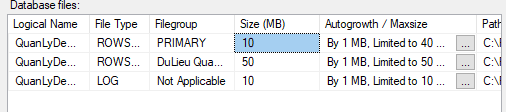
ALTER DATABASE QuanLyDeAn

SET READ\_WRITE

**  
g. Dùng lệnh Alter DataBase … MODIFY FILE … để tăng SIZE của QuanLyDeAn  
\_data1 thành 50 MB. Tương tự tăng SIZE của tập tin QuanLyDeAn \_log thành  
10 MB.**

****

****

****

Module 2

1. **Cho CSDL Quản lý đề án, hãy tạo CSDL có tên\_QuanLyDeAn\_TenSV trong thư mục tùy ý  
   . Sau đó, thiết kế cấu trúc các bảng có khóa chính khóa ngoại và kiểu dữ liệu theo lược  
   đồ sau (thực hiện bằng DDL)**

**Đầu tiên tạo ra Database tên :** QuanLyDeAn\_TenSV

CREATE DATABASE QuanLyDeAn\_TenSV

ON

( NAME = 'QuanLyDeAn\_TenSV\_dat',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.THIEN\MSSQL\DATA\Thien\QuanLyDeAn\_TenSV.mdf',

SIZE = 10,

MAXSIZE = 50,

FILEGROWTH = 5 )

LOG ON

( NAME = 'QuanLyDeAn\_TenSV\_log',

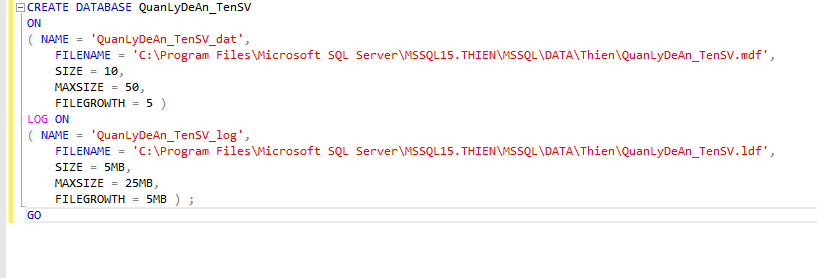
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.THIEN\MSSQL\DATA\Thien\QuanLyDeAn\_TenSV.ldf',

SIZE = 5MB,

MAXSIZE = 25MB,

FILEGROWTH = 5MB ) ;

GO

****

TẠO BẢNG NHANVIEN :

Create table NHANVIEN

(

MANV varchar(9) NOT NULL Primary Key,

HONV nvarchar(15) ,

TENLOT NVARCHAR(30),

TENNV NVARCHAR(30),

NGSINH SMALLDATETIME,

DCHI NVARCHAR(150),

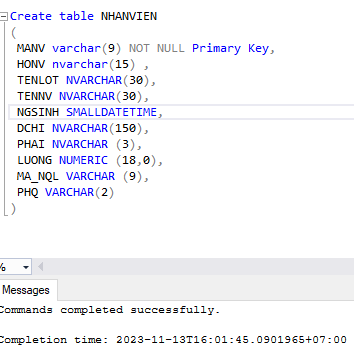
PHAI NVARCHAR (3),

LUONG NUMERIC (18,0),

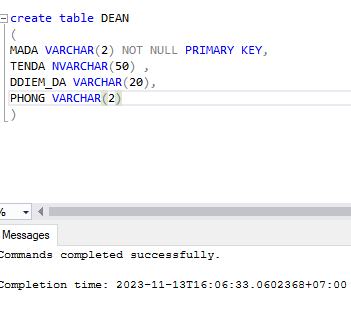
MA\_NQL VARCHAR (9),

PHQ VARCHAR(2)

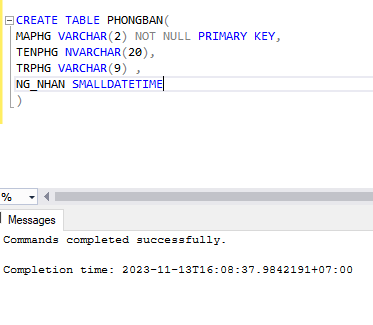
)

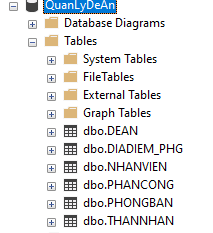


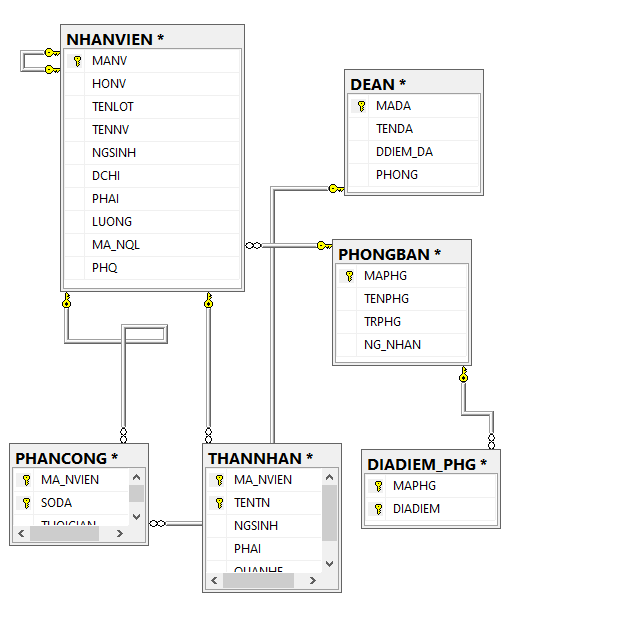
TẠO BẢNG DEAN



TẠO BẢNG PHONGBAN

…





**2. Tìm hiểu về kiểu dữ liệu (datatype):**  
**a. Tìm hiểu và trả lời các câu hỏi sau:  
- Có mấy loại datatype, hãy liệt kê.**

Số chính xác (Exact numeric): bit, tinyint, smallint, int, bigint, decimal, numeric, money và smallmoney.

Số xấp xỉ (Approximate numeric): float và real.

Ngày và thời gian (Date and time): date, datetime, datetime2, datetimeoffset, smalldatetime, time.

Chuỗi ký tự (Character strings): char, varchar, text.

**- Các system datatype được SQL Server lưu trữ trong Table nào ở trong  
CSDL nào.**

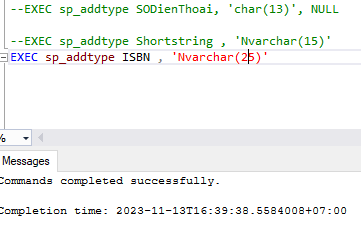
Các kiểu dữ liệu hệ thống (system datatype) trong SQL Server được lưu trữ trong bảng sys.types

**- Các User-defined datatype được SQL Server lưu trữ trong Table nào ở  
trong CSDL nào?**

Các kiểu dữ liệu do người dùng định nghĩa (User-defined datatypes) trong SQL Server được lưu trữ trong bảng sys.types  
**b. Vào màn hình Query Editor, chọn QuanLyDeAn là CSDL hiện hành, định nghĩa**các datatype:

|  |  |
| --- | --- |
| **KIỂU DỮ LIỆU (DATA TYPE)** | **MÔ TẢ DỮ LIỆU (DESCRIPTION OF DATA)** |
| ISBN | 25 ký tự |
| SoDienThoai | 13 ký tự , chấp nhận NULL |
| Shortstring | Số ký tự thay đổi đến 15 ký tự |

**Dùng thủ tục sp\_addtype hoặc Create Type để định nghĩa  
Ví dụ: EXEC sp\_addtype SODienThoai, 'char(13)', NULL**

**  
c. Các User-defined datatype vừa định nghĩa được lưu trữ ở đâu và phạm vi sử dụng của nó ở đâu (trong toàn bộ một instance hay chỉ ở trong CSDL hiện hành).**

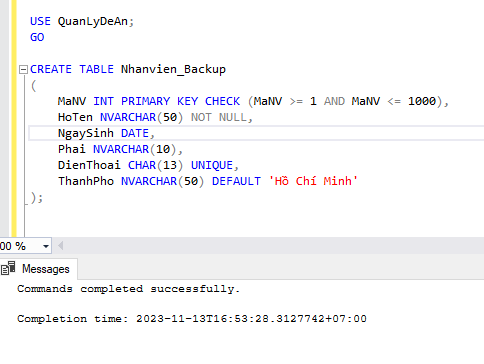
Các kiểu dữ liệu do người dùng định nghĩa (User-defined datatypes) trong SQL Server được lưu trữ trong bảng sys.types  
**d. Có bao nhiêu cách liệt kê danh sách các các User-Defined datatype vừa định nghĩa.**

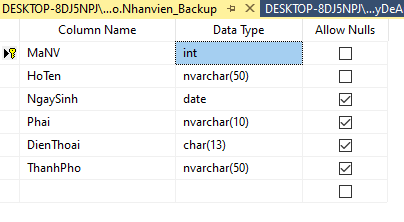
Sử dụng INFORMATION\_SCHEMA.DOMAINS và sử dụng bảng sys.types  
**SELECT domain\_name, data\_type, character\_maximum\_length  
FROM information\_schema.domains  
ORDER BY domain\_name  
Hoặc SELECT \* From Systype  
e. Muốn User-Defined datatype được dùng trong tất cả các CSDL thì bạn định nghĩa nó ở đâu?**

Để một kiểu dữ liệu do người dùng định nghĩa (User-Defined datatype) có thể được sử dụng trong tất cả các cơ sở dữ liệu, cần định nghĩa nó ở mức server chứ không phải ở mức cơ sở dữ liệu hoặc bằng cách sử dụng câu lệnh SQL CREATE TYPE

**f. Bạn hãy tạo 1 bảng có tên là Nhanvien\_Backup(MaNV, Hoten, NgaySinh, Phai,DienThoai, ThanhPho) trong CSDL QuanLyDeAn và sử dụng User-defined datatype vừa định nghĩa ở trên. Trong đó, thêm các ràng buộc sau:**

- MaNV có giá trị từ 1🡪1000, Primary Key  
- HoTen không được bỏ trống  
- DienThoai có 13 số, và duy nhất (Unique)  
- ThanhPho mặc định là Hồ Chí Minh. (default)



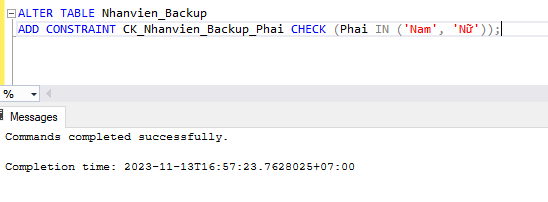
  
**g. Hãy xóa kiểu dữ liệu SoDienThoai. Có xóa được hay không? Tại sao?**

Xóa được vì đó là kiểu dữ liệu người dùng từ định nghĩa

**3. Cài đặt thêm các ràng buộc check, rule, default:**  
**a. Hãy tạo ràng buộc cho cột Phai của bảng Nhanvien\_Backup chỉ có “Nam” hoặc “Nữ”**

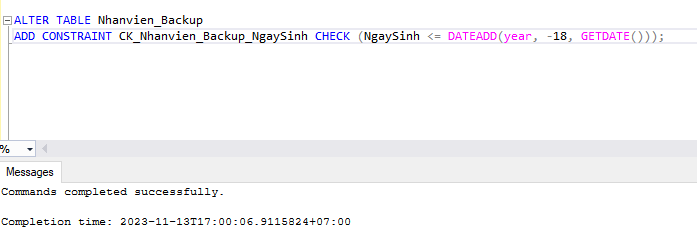
ALTER TABLE Nhanvien\_Backup

ADD CONSTRAINT CK\_Nhanvien\_Backup\_Phai CHECK (Phai IN ('Nam', 'Nữ'));

**  
b. Hãy tạo ràng buộc cho dữ liệu cho cột NgaySinh <=Ngayhientai-18 (năm) trong bảng Nhanvien\_Backup.**

ALTER TABLE Nhanvien\_Backup

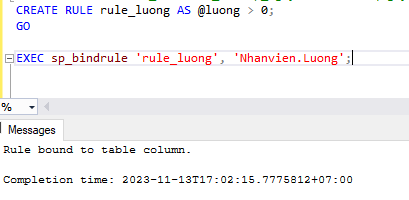
ADD CONSTRAINT CK\_Nhanvien\_Backup\_NgaySinh CHECK (NgaySinh <= DATEADD(year, -18, GETDATE()));

**  
c. Tạo 1 rule có tên là rule\_luong cho phép chứa giá trị lớn hơn 0, rồi gắn nó vào cột Luong trong bảng Nhanvien.**

CREATE RULE rule\_luong AS @luong > 0;

GO

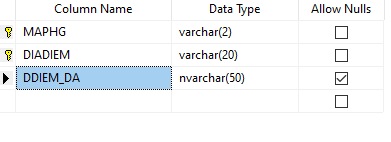
EXEC sp\_bindrule 'rule\_luong', 'Nhanvien.Luong';

**  
d. Tạo 1 default có giá trị là ‘Tp.Hồ Chí Minh’ , sau đó gắn vào cột DDIEM\_DA của bảng DIADIEM.**

CREATE DEFAULT Default\_DDIEM\_DA AS 'Tp.Hồ Chí Minh';

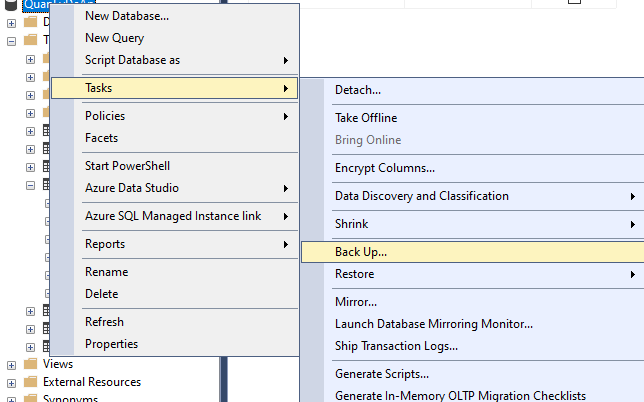
GO

EXEC sp\_bindefault 'Default\_DDIEM\_DA', 'DIADIEM.DDIEM\_DA';

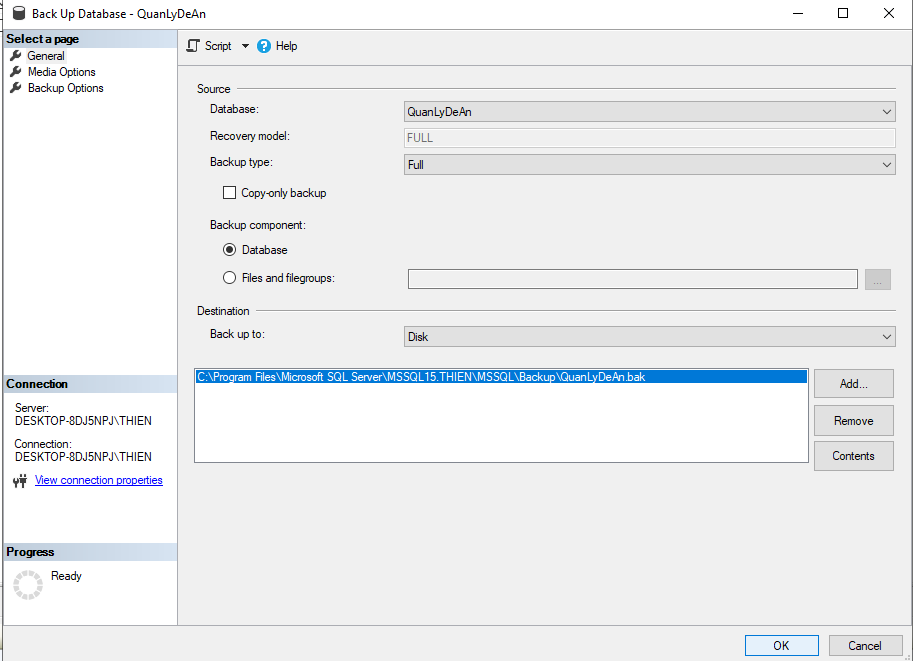


**4. Hãy thực hiện việc Backup và Retore CSDL Quanlydean.**

**Chọn chuột phải 🡪 taks 🡪 backup**

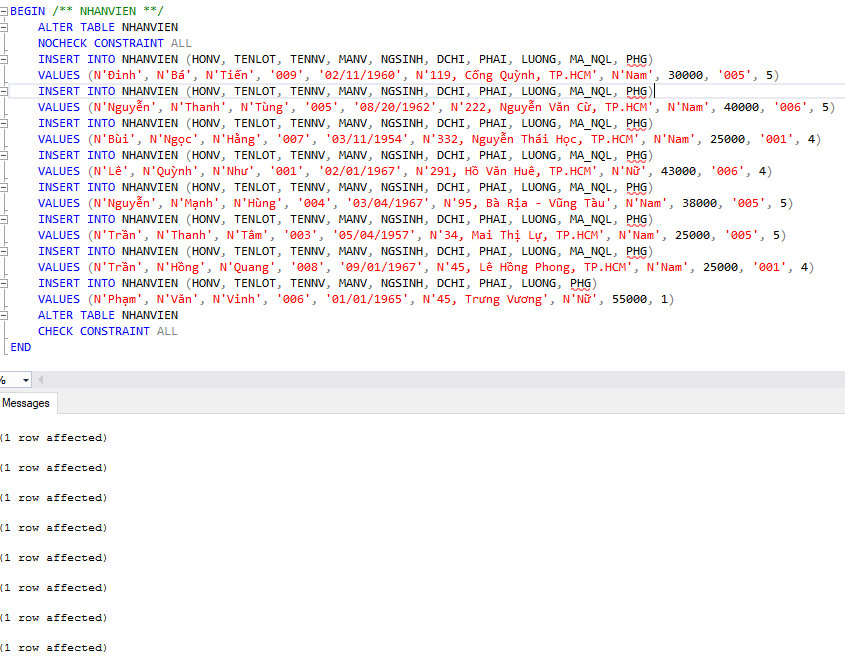


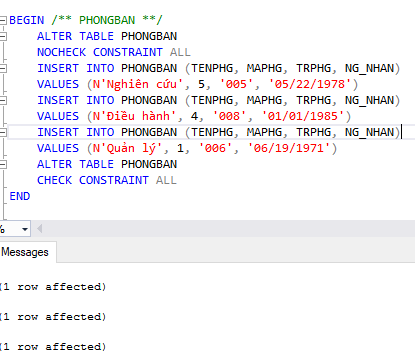
Chọn đường dẫn để Backup

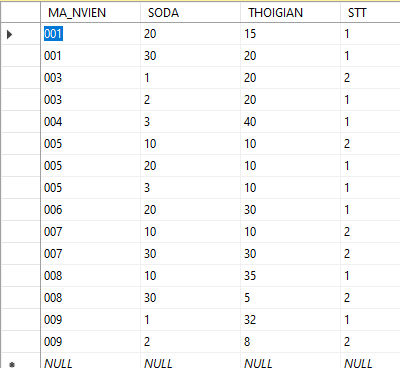


**Module 3:**

**Phần 1: câu lệnh INSERT**1. Thực hiện việc attach để attach CSDL QuanLyDeAn được GV cung cấp tại phòng  
thực hành. Sau đó, chèn dữ liệu vào các bảng theo mẫu lệnh sau. Kiểm tra dữ liệu  
sau mỗi lần thực hiện.  
INSERT table\_name  
SELECT \* FROM table\_name





  
2. Hiệu chỉnh kiểu dữ liệu cho cột TENPHG thành nvarchar(20). Thêm vào bảng  
PhongBan dòng sau:  
**2 Nhân sự 10 2003/12/12**🡪 kiểm tra có lỗi gì xảy ra?. Bạn phải làm gì?

🡪 KHÔNG CÓ LỖI NÀO XẢY RA

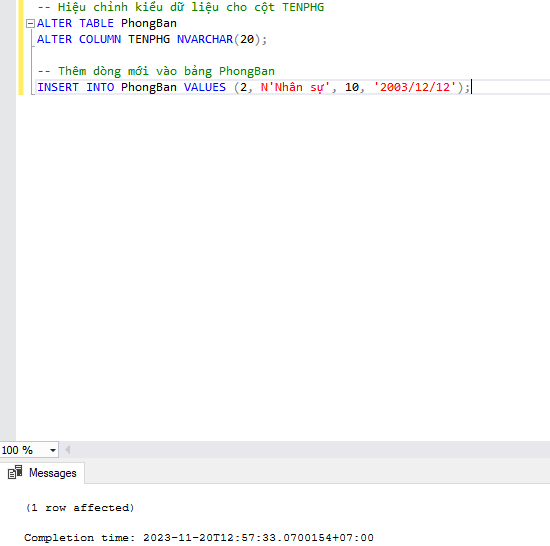
-- Hiệu chỉnh kiểu dữ liệu cho cột TENPHG

ALTER TABLE PhongBan

ALTER COLUMN TENPHG NVARCHAR(20);

-- Thêm dòng mới vào bảng PhongBan

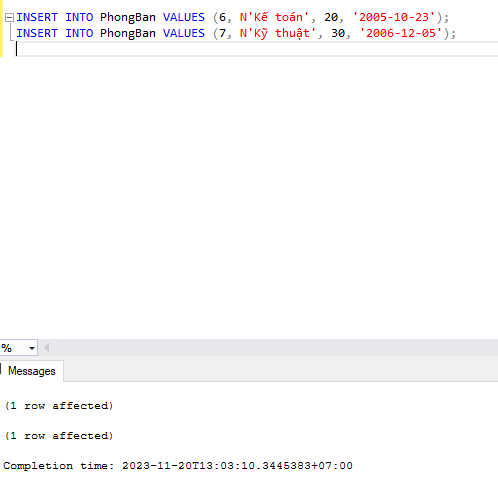
INSERT INTO PhongBan VALUES (2, N'Nhân sự', 10, '2003/12/12');

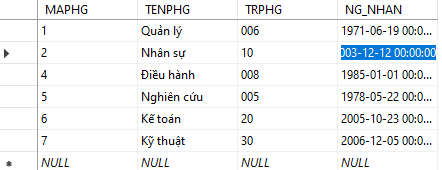
  
3. Thực hiện thêm tiếp các record sau:

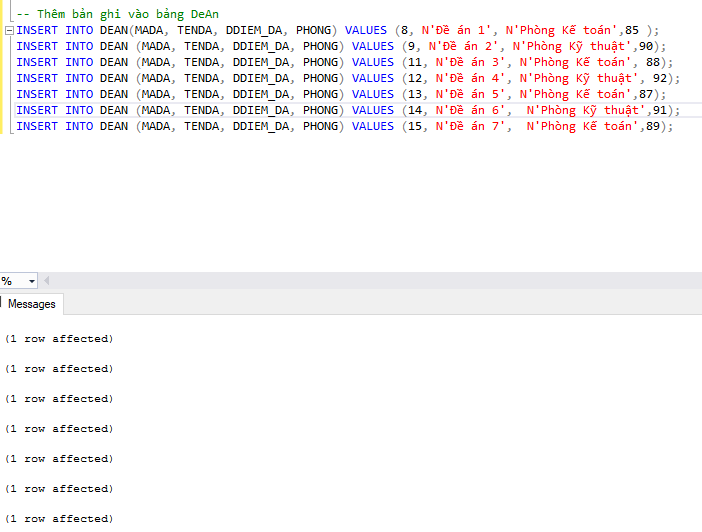


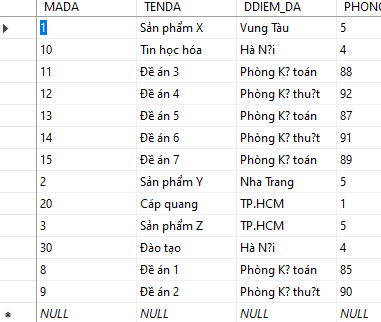
INSERT INTO PhongBan VALUES (6, N'Kế toán', 20, '2005-10-23');

INSERT INTO PhongBan VALUES (7, N'Kỹ thuật', 30, '2006-12-05');



  
4. Thêm tùy ý vào bảng DeAn các record có mã đề án (MADA) là  
8,9,11,12,13,14,15. Dữ liệu các cột còn lại tùy chọn.





5. Thêm vào bảng phân công (PHANCONG) 50 dòng cho các nhân viên có mã số từ  
10🡪50 tương ứng với các đề án 8,9,11,12,13,14,15 (nhập bằng cửa sổ Edit top  
200 rows, không cần dùng lệnh).

BEGIN /\*\* PHANCONG \*\*/

ALTER TABLE PHANCONG

NOCHECK CONSTRAINT ALL

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('036', 2, 1, 32)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('037', 3, 2, 8)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('038', 4, 1, 40)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('039', 5, 2, 20.0)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('040', 6, 1, 20.0)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('041', 11, 1, 35)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('042', 12, 2, 5)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('043', 13, 1, 20)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('044', 14, 1, 15)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('045', 15, 1, 30)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('046',17, 1, 10)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('047', 18, 2, 10)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('048', 21, 1, 10)

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('049', 30, 2, 30)

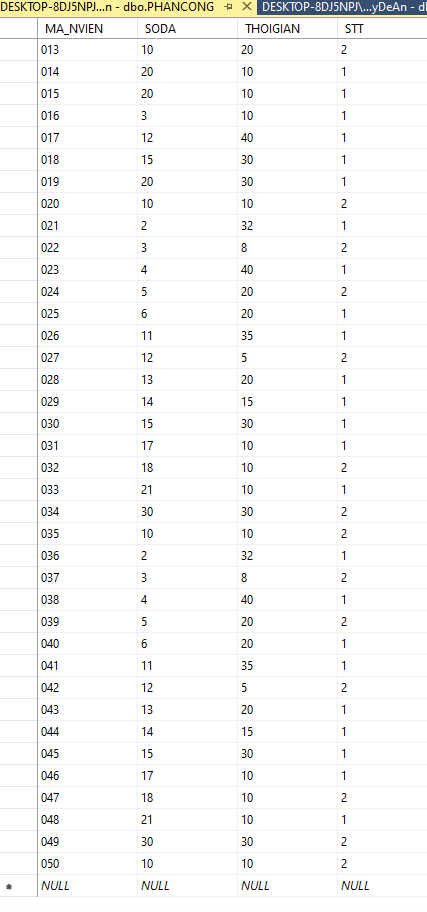
INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('050', 10, 2, 10)

ALTER TABLE PHANCONG

CHECK CONSTRAINT ALL

END



**Phần 2: câu lệnh UPDATE**1. Cập nhật địa chỉ cho các nhân viên có địa chỉ =null là Tp.Hồ Chí Minh trong table Nhanvien

update NHANVIEN Set DCHI = 'TP.Ho Chi Minh' where DCHI is null;



2. Cập nhật PHG =2 cho những nhân viên có mã số từ 1 đến 10; PHG=6 cho các  
nhân viên có mã số từ 11 đến 40, PHG=7 cho các nhân viên có mã số từ 41 đến  
80, PHG =5 cho các nhân viên có mã số từ 102🡪200.

update NHANVIEN set PHG = case

when MANV between 1 and 10 then 2

when MANV between 11 and 40 then 6

when MANV between 41 and 80 then 7

when MANV between 102 and 200 then 5

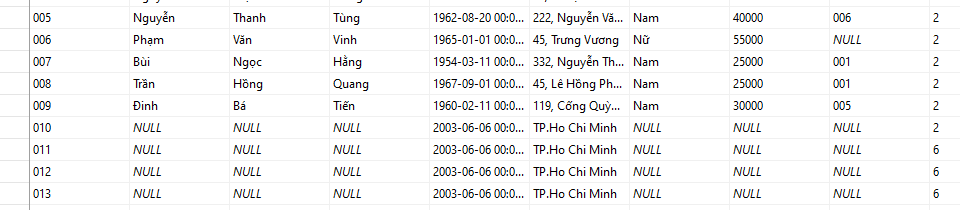
else PHG

end;



3. Cập nhật ngày sinh cho những nhân viên có ngày sinh = null (is null) là ngày  
01/01/1985 trong table Nhanvien.

update NHANVIEN set NGSINH = '2003-06-06' where NGSINH is null;



4. Cập nhật LUONG =100000 cho các nhân viên ở phòng NghienCuu, LUONG  
=65000 cho các nhân viên ở phòng Kế toán, LUONG=85000 cho các nhân viên ở  
phòng Kỹ thuật, LUONG = 70000 cho các nhân viên ở phòng nhân sự.

update NHANVIEN set LUONG = case

when PHONGBAN.TENPHG = 'Nghiên cứu' then 100000

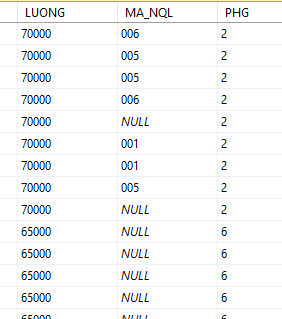
when PHONGBAN.TENPHG = 'Kế toán' then 65000

when PHONGBAN.TENPHG = 'Kỹ thuật' then 85000

when PHONGBAN.TENPHG = 'Nhân sự' then 7000

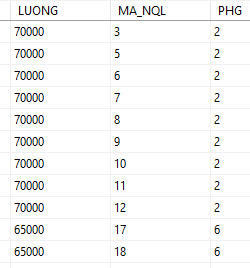
else LUONG

end from NHANVIEN join PHONGBAN on NHANVIEN.PHG = PHONGBAN.MAPHG;

   
5. Cập nhật mã người quản lý (MA\_NQL) trong bảng nhân viên tương ứng với  
phòng ban mà nhân viên đó thuộc về.  
VDnhân viên có mã nv =8, thuộc phòng ban 2 thì mã người quản lý là 10.

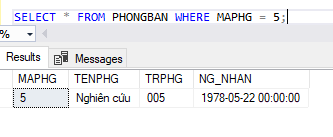
UPDATE NHANVIEN

SET MA\_NQL = CAST(MANV AS INT) + CAST(PHG AS INT);

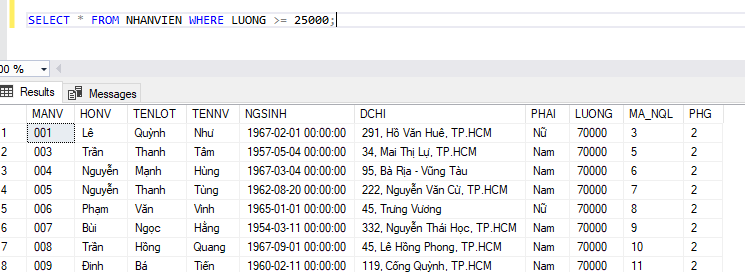


**Phần 3: câu lệnh SELECT  
3.1 TRUY VẤN CƠ BẢN  
3.1.1 Truy vấn đơn giản**1. Tìm các nhân viên làm việc ở phòng số 5

SELECT \* FROM PHONGBAN WHERE MAPHG = 5;

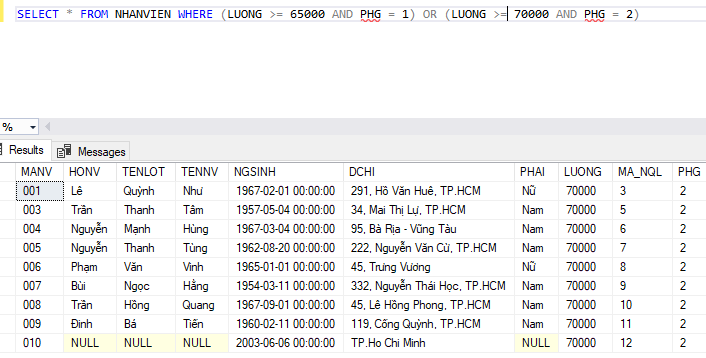
  
2. Tìm các nhân viên có mức lương trên 25000

SELECT \* FROM NHANVIEN WHERE LUONG >= 25000;



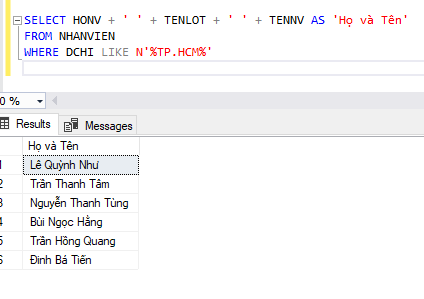
3. Tìm các nhân viên có mức lương trên 65,000 ở phòng 1 hoặc các nhân viên có  
mức lương trên 70,000 ở phòng 2

SELECT \* FROM NHANVIEN WHERE (LUONG >= 65000 AND PHG = 1) OR (LUONG >= 70000 AND PHG = 2)

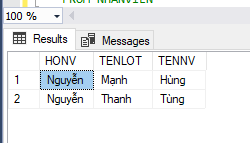


4. Cho biết họ tên đầy đủ của các nhân viên ở TP HCM

SELECT HONV + ' ' + TENLOT + ' ' + TENNV AS 'Họ và Tên' FROM NHANVIEN WHERE DCHI LIKE N'%TP.HCM%'

  
5. Cho biết họ tên đầy đủ của các nhân viên có họ bắt đầu bằng ký tự 'N'

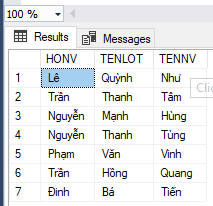
SELECT HONV, TENLOT, TENNV FROM NHANVIEN WHERE HONV LIKE 'N%'

  
6. Cho biết ngày sinh và địa chỉ của nhân viên Lê Minh Tính

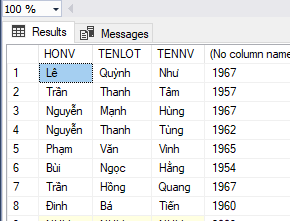
SELECT NGSINH,DCHI FROM NHANVIEN WHERE HONV = N'Lê' AND TENLOT = N'Minh' AND TENNV = N'Tính'

7. Cho biết các nhân viên có năm sinh trong khoảng 1955 đến 1975

SELECT HONV, TENLOT, TENNV FROM NHANVIEN WHERE NGSINH BETWEEN '1955' AND '1975'

  
8. Cho biết các nhân viên và năm sinh của nhân viên

SELECT HONV, TENLOT, TENNV, YEAR(NGSINH) FROM NHANVIEN

  
9. Cho biết các nhân viên và tuổi của nhân viên

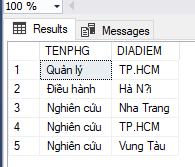
SELECT HONV,TENLOT,TENNV ,YEAR('2023/11/20') - YEAR(NGSINH) AS Tuoi

FROM NHANVIEN

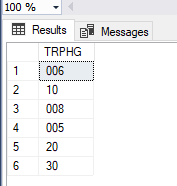
  
**3.1.2 Truy vấn có sử dụng phép kết**10.Với mỗi phòng ban, cho biết tên phòng ban và địa điểm phòng

SELECT PHONGBAN.TENPHG, DIADIEM\_PHG.DIADIEM FROM PHONGBAN

INNER JOIN DIADIEM\_PHG ON PHONGBAN.MAPHG = DIADIEM\_PHG.MAPHG

  
11.Tìm tên những người trưởng phòng của từng phòng ban

SELECT TRPHG FROM PHONGBAN

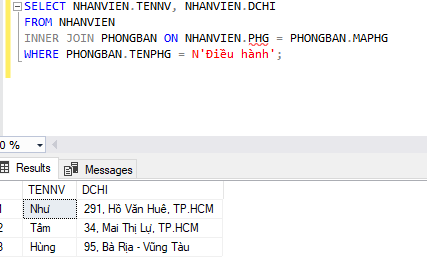
  
12.Tìm tên và địa chỉ của tất cả các nhân viên của phòng "Điều hành".

SELECT NHANVIEN.TENNV, NHANVIEN.DCHI

FROM NHANVIEN

INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

WHERE PHONGBAN.TENPHG = N'Điều hành';

  
13.Với mỗi đề án ở Nha TRang, cho biết tên đề án, tên phòng ban, họ tên và ngày  
nhận chức của trưởng phòng của phòng ban chủ trì đề án đó.

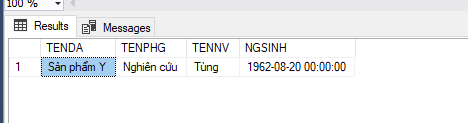
SELECT DEAN.TENDA, PHONGBAN.TENPHG, NHANVIEN.TENNV, NHANVIEN.NGSINH

FROM DEAN

INNER JOIN PHONGBAN ON DEAN.PHONG = PHONGBAN.MAPHG

INNER JOIN NHANVIEN ON PHONGBAN.TRPHG = NHANVIEN.MANV

WHERE DEAN.DDIEM\_DA = N'Nha Trang';

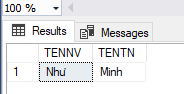
  
14.Tìm tên những nữ nhân viên và tên người thân của họ

SELECT NHANVIEN.TENNV, THANNHAN.TENTN

FROM NHANVIEN

INNER JOIN THANNHAN ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

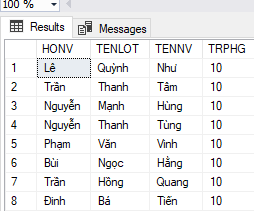
WHERE NHANVIEN.PHAI = N'Nữ';

  
15.Với mỗi nhân viên, cho biết họ tên nhân viên và họ tên người quản lý trực tiếp  
của nhân viên đó

SELECT NHANVIEN.HONV , NHANVIEN.TENLOT,NHANVIEN.TENNV , PHONGBAN.TRPHG

FROM NHANVIEN

INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

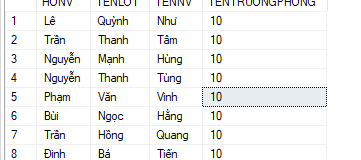


16.Với mỗi nhân viên, cho biết họ tên của nhân viên đó, họ tên người trưởng phòng và họ tên người quản lý trực tiếp của nhân viên đó.

SELECT NHANVIEN.HONV , NHANVIEN.TENLOT,NHANVIEN.TENNV , PHONGBAN.TRPHG as TENTRUONGPHONG

FROM NHANVIEN

INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

  
17.Tên những nhân viên phòng số 5 có tham gia vào đề án "Xây dựng nhà máy chế  
biến thủy sản" và tên người quản lý trực tiếp.

SELECT HONV, TENLOT,TENNV, TRPHG FROM NHANVIEN

INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

INNER JOIN DEAN ON PHONGBAN.MAPHG = DEAN.PHONG

WHERE PHG = 5 AND TENDA = N'Xây dựng nhà máy chế biến thủy sản'  
18.Cho biết tên các đề án mà nhân viên Trần Anh Tuấn đã tham gia.

SELECT NHANVIEN.HONV ,NHANVIEN.TENLOT ,NHANVIEN.TENNV ,DEAN.TENDA

FROM DEAN

INNER JOIN PHANCONG ON DEAN.MADA = PHANCONG.SODA

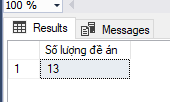
INNER JOIN NHANVIEN ON PHANCONG.MA\_NVIEN = NHANVIEN.MANV

WHERE NHANVIEN.HONV = N'Trần'AND NHANVIEN.TENLOT = N'Anh' AND NHANVIEN.TENNV = N'Tuấn'

**3.2 GOM NHÓM**19.Cho biết số lượng đề án của công ty

SELECT COUNT(\*) AS 'Số lượng đề án'

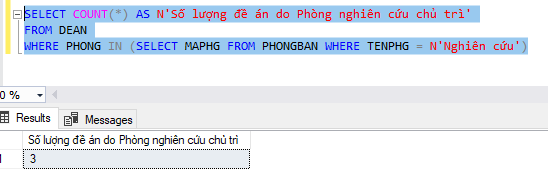
FROM DEAN;

  
20.Cho biết số lượng đề án do phòng 'Nghiên Cứu' chủ trì

SELECT COUNT(\*) AS N'Số lượng đề án do Phòng nghiên cứu chủ trì'

FROM DEAN

WHERE PHONG IN (SELECT MAPHG FROM PHONGBAN WHERE TENPHG = N'Nghiên cứu')

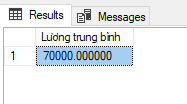


21.Cho biết lương trung bình của các nữ nhân viên

SELECT AVG(LUONG) AS 'Lương trung bình'

FROM NHANVIEN

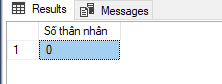
WHERE PHAI = N'Nữ'

  
22.Cho biết số thân nhân của nhân viên ‘Lê Thanh Sang'

SELECT COUNT(\*) AS 'Số thân nhân'

FROM NHANVIEN, THANNHAN

WHERE NHANVIEN.MANV = THANNHAN.MA\_NVIEN AND NHANVIEN.HONV = N'Lê' AND NHANVIEN.TENLOT = N'Thanh' AND NHANVIEN.TENNV = N'Sang'

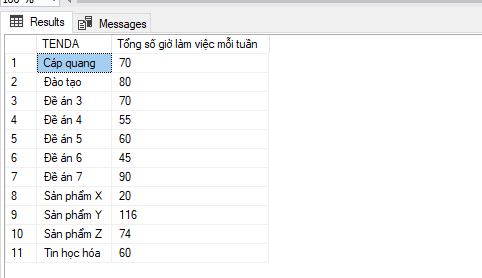
  
23.Với mỗi đề án, liệt kê tên đề án và tổng số giờ làm việc một tuần của tất cả các  
nhân viên tham dự đề án đó.

SELECT DEAN.TENDA, SUM(PHANCONG.THOIGIAN) AS 'Tổng số giờ làm việc mỗi tuần'

FROM DEAN, PHANCONG

WHERE DEAN.MADA = PHANCONG.SODA

GROUP BY DEAN.TENDA

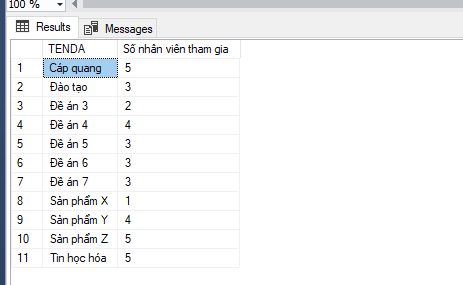
  
24.Với mỗi đề án, cho biết có bao nhiêu nhân viên tham gia đề án đó

SELECT DEAN.TENDA, COUNT(DISTINCT PHANCONG.MA\_NVIEN) AS 'Số nhân viên tham gia'

FROM DEAN, PHANCONG

WHERE DEAN.MADA = PHANCONG.SODA

GROUP BY DEAN.TENDA

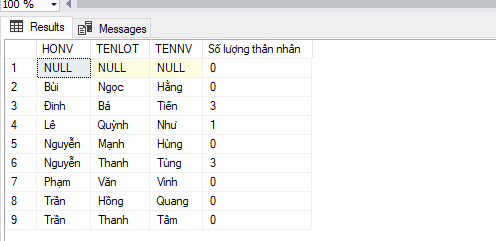
  
25.Với mỗi nhân viên, cho biết họ và tên nhân viên và số lượng thân nhân của nhân viên đó.

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV, COUNT(THANNHAN.MA\_NVIEN) AS 'Số lượng thân nhân'

FROM NHANVIEN LEFT JOIN THANNHAN

ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

GROUP BY NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENN

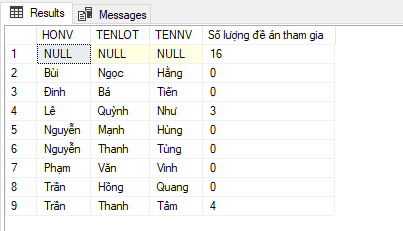
  
26.Với mỗi nhân viên, cho biết họ tên của nhân viên và số lượng đề án mà nhân  
viên đó đã tham gia.

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV, COUNT(DISTINCT PHANCONG.SODA) AS 'Số lượng đề án tham gia'

FROM NHANVIEN LEFT JOIN PHANCONG

ON NHANVIEN.MANV = PHANCONG.MA\_NVIEN

GROUP BY NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

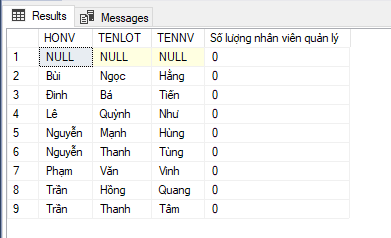
  
27.Với mỗi nhân viên, cho biết số lượng nhân viên mà nhân viên đó quản lý trực  
tiếp.

SELECT N1.HONV, N1.TENLOT, N1.TENNV, COUNT(N2.MANV) AS 'Số lượng nhân viên quản lý'

FROM NHANVIEN N1 LEFT JOIN NHANVIEN N2

ON N1.MANV = N2.MA\_NQL

GROUP BY N1.HONV, N1.TENLOT, N1.TENNV

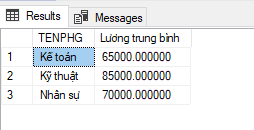
  
28.Với mỗi phòng ban, liệt kê tên phòng ban và lương trung bình của những nhân  
viên làm việc cho phòng ban đó.

SELECT PHONGBAN.TENPHG, AVG(NHANVIEN.LUONG) AS 'Lương trung bình'

FROM PHONGBAN, NHANVIEN

WHERE PHONGBAN.MAPHG = NHANVIEN.PHG

GROUP BY PHONGBAN.TENPHG

  
29.Với các phòng ban có mức lương trung bình trên 30,000, liệt kê tên phòng ban  
và số lượng nhân viên của phòng ban đó.

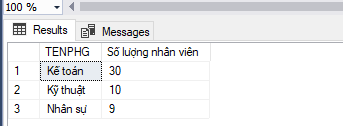
SELECT PHONGBAN.TENPHG, COUNT(NHANVIEN.MANV) AS 'Số lượng nhân viên'

FROM PHONGBAN JOIN NHANVIEN

ON PHONGBAN.MAPHG = NHANVIEN.PHG

GROUP BY PHONGBAN.TENPHG

HAVING AVG(NHANVIEN.LUONG) > 30000

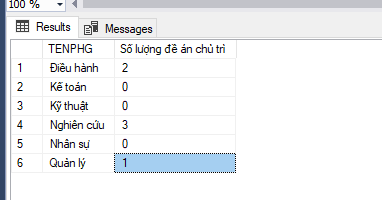
  
30.Với mỗi phòng ban, cho biết tên phòng ban và số lượng đề án mà phòng ban đó  
chủ trì

SELECT PHONGBAN.TENPHG, COUNT(DEAN.MADA) AS 'Số lượng đề án chủ trì'

FROM PHONGBAN LEFT JOIN DEAN

ON PHONGBAN.MAPHG = DEAN.PHONG

GROUP BY PHONGBAN.TENPHG

  
31.Với mỗi phòng ban, cho biết tên phòng ban, họ tên người trưởng phòng và số  
lượng đề án mà phòng ban đó chủ trì

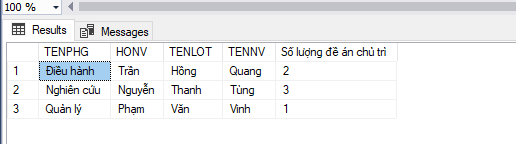
SELECT PHONGBAN.TENPHG, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV, COUNT(DEAN.MADA) AS 'Số lượng đề án chủ trì'

FROM PHONGBAN

JOIN NHANVIEN ON PHONGBAN.TRPHG = NHANVIEN.MANV

LEFT JOIN DEAN ON PHONGBAN.MAPHG = DEAN.PHONG

GROUP BY PHONGBAN.TENPHG, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

  
32.Với mỗi phòng ban có mức lương trung bình lớn hơn 50,000, cho biết tên phòng ban và số lượng đề án mà phòng ban đó chủ trì.

SELECT PHONGBAN.TENPHG, COUNT(DEAN.MADA) AS 'Số lượng đề án chủ trì'

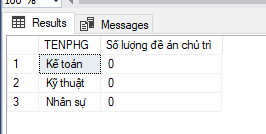
FROM PHONGBAN

JOIN NHANVIEN ON PHONGBAN.MAPHG = NHANVIEN.PHG

LEFT JOIN DEAN ON PHONGBAN.MAPHG = DEAN.PHONG

GROUP BY PHONGBAN.TENPHG

HAVING AVG(NHANVIEN.LUONG) > 50000

  
33.Cho biết số đề án diễn ra tại từng địa điểm

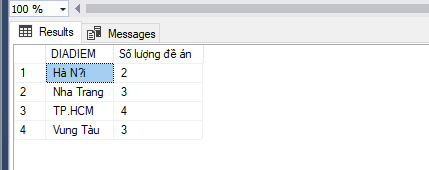
SELECT DIADIEM\_PHG.DIADIEM, COUNT(DEAN.MADA) AS 'Số lượng đề án'

FROM DIADIEM\_PHG

JOIN PHONGBAN ON DIADIEM\_PHG.MAPHG = PHONGBAN.MAPHG

JOIN DEAN ON PHONGBAN.MAPHG = DEAN.PHONG

GROUP BY DIADIEM\_PHG.DIADIEM

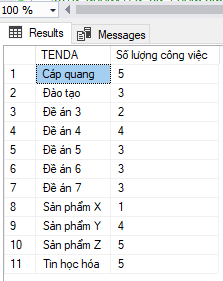
  
34.Với mỗi đề án, cho biết tên đề án và số lượng công việc của đề án này.

SELECT DEAN.TENDA, COUNT(PHANCONG.MA\_NVIEN) AS 'Số lượng công việc'

FROM DEAN

JOIN PHANCONG ON DEAN.MADA = PHANCONG.SODA

GROUP BY DEAN.TENDA



**3.3 TRUY VẤN LỒNG + GOM NHÓM**36.Cho biết danh sách các đề án (MADA) có: nhân công với họ (HONV) là 'Lê' hoặc  
có người trưởng phòng chủ trì đề án với họ (HONV) là 'Lê'.

SELECT DEAN.MADA AS 'Mã đề án'

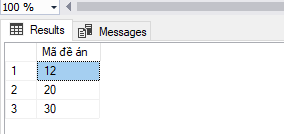
FROM DEAN

JOIN PHANCONG ON DEAN.MADA = PHANCONG.SODA

JOIN NHANVIEN ON PHANCONG.MA\_NVIEN = NHANVIEN.MANV OR DEAN.PHONG = NHANVIEN.PHG

WHERE NHANVIEN.HONV = N'Lê'

GROUP BY DEAN.MADA

  
37.Danh sách những nhân viên (HONV, TENLOT, TENNV) có trên 2 thân nhân.

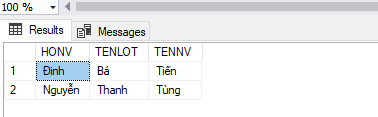
SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

FROM NHANVIEN

JOIN THANNHAN ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

GROUP BY NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

HAVING COUNT(THANNHAN.MA\_NVIEN) > 2

  
38.Danh sách những nhân viên (HONV, TENLOT, TENNV) không có thân nhân nào  
(dùng NOT IN, LEFT JOIN, NOT EXISTS)

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

FROM NHANVIEN

WHERE NHANVIEN.MANV NOT IN (SELECT THANNHAN.MA\_NVIEN FROM THANNHAN)

Dùng LEFT JOIN

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

FROM NHANVIEN

LEFT JOIN THANNHAN ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

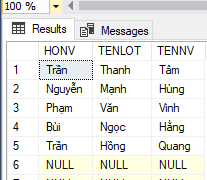
WHERE THANNHAN.MA\_NVIEN IS NULL

DÙNG NOT EXISTS

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

FROM NHANVIEN

WHERE NOT EXISTS (SELECT 1 FROM THANNHAN WHERE NHANVIEN.MANV = THANNHAN.MA\_NVIEN)

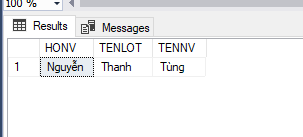
  
39.Danh sách những trưởng phòng (HONV, TENLOT, TENNV) có tối thiểu một than nhân.

SELECT NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

FROM NHANVIEN

JOIN PHONGBAN ON NHANVIEN.MANV = PHONGBAN.TRPHG

WHERE EXISTS (SELECT 1 FROM THANNHAN WHERE NHANVIEN.MANV = THANNHAN.MA\_NVIEN)

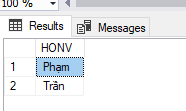
  
40.Tìm họ (HONV) của những trưởng phòng chưa có gia đình.

SELECT NHANVIEN.HONV

FROM NHANVIEN

JOIN PHONGBAN ON NHANVIEN.MANV = PHONGBAN.TRPHG

WHERE NOT EXISTS (SELECT 1 FROM THANNHAN WHERE NHANVIEN.MANV = THANNHAN.MA\_NVIEN)

  
41.Cho biết họ tên nhân viên (HONV, TENLOT, TENNV) có mức lương trên mức  
lương trung bình của phòng "Nhân sự"

SELECT N1.HONV, N1.TENLOT, N1.TENNV

FROM NHANVIEN N1

WHERE N1.LUONG > (

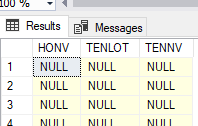
SELECT AVG(N2.LUONG)

FROM NHANVIEN N2

JOIN PHONGBAN ON N2.PHG = PHONGBAN.MAPHG

WHERE PHONGBAN.TENPHG = N'Nhân sự'

)

  
42.Cho biết tên phòng ban và họ tên trưởng phòng của phòng ban có đông nhân  
viên nhất.

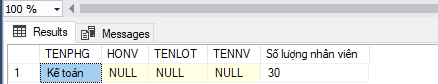
SELECT TOP 1 PHONGBAN.TENPHG, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV, COUNT(\*) AS 'Số lượng nhân viên'

FROM PHONGBAN

JOIN NHANVIEN ON PHONGBAN.MAPHG = NHANVIEN.PHG

GROUP BY PHONGBAN.TENPHG, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

ORDER BY COUNT(\*) DESC

  
43.Cho biết danh sách các mã đề án mà nhân viên có mã là 60 chưa làm.

SELECT DEAN.MADA

FROM DEAN

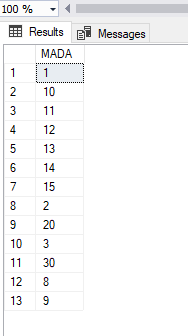
WHERE DEAN.MADA NOT IN (

SELECT PHANCONG.SODA

FROM PHANCONG

WHERE PHANCONG.MA\_NVIEN = '60'

)



44.Danh sách nhân viên gồm MaNv, HovaTen (được nối từ các cột HONV, TENLOT,  
TENNV) và địa chỉ (DCHI) của những nhân viên làm việc cho một đề án ở 'TP  
HCM' nhưng phòng ban mà họ trực thuộc lại không tọa lạc ở thành phố 'TP HCM'  
.

SELECT NHANVIEN.MANV 'Mã nhân viên', CONCAT(NHANVIEN.HONV, ' ', NHANVIEN.TENLOT, ' ', NHANVIEN.TENNV) AS 'Họ và Tên', NHANVIEN.DCHI AS 'Địa chỉ'

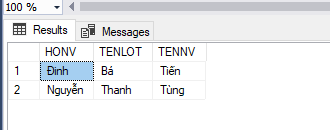
FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MANV = PHANCONG.MA\_NVIEN

JOIN DEAN ON PHANCONG.SODA = DEAN.MADA

JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

JOIN DIADIEM\_PHG ON PHONGBAN.MAPHG = DIADIEM\_PHG.MAPHG

WHERE DEAN.DDIEM\_DA = 'TP.HCM' AND DIADIEM\_PHG.DIADIEM <> 'TP.HCM'  
45. Tổng quát câu 16, tìm họ tên và địa chỉ của các nhân viên làm việc cho một đề án ở một thành phố nhưng phòng ban mà họ trực thuộc lại không toạ lạc ở thành  
phố đó.

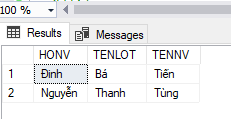
SELECT NHANVIEN.HONV 'Họ', NHANVIEN.TENLOT 'Tên lót', NHANVIEN.TENNV 'Tên', NHANVIEN.DCHI 'Địa chỉ'

FROM NHANVIEN

INNER JOIN DEAN ON NHANVIEN.PHG = DEAN.PHONG

INNER JOIN DIADIEM\_PHG ON DEAN.PHONG = DIADIEM\_PHG.MAPHG

WHERE DEAN.DDIEM\_DA != DIADIEM\_PHG.DIADIEM

  
**3.5 COMPUTE/COMPUTE BY**46.Danh sách nhân viên gồm MaNv, HovaTen (được nối từ các cột HONV, TENLOT,  
TENNV), Tuổi và tuổi trung bình của các nhân viên trong công ty (dùng  
COMPUTE)

SELECT

MANV,

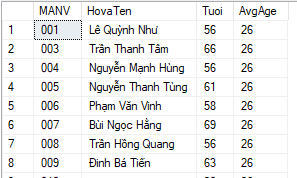
CONCAT(HONV, ' ', TENLOT, ' ', TENNV) AS HovaTen,

YEAR(CURRENT\_TIMESTAMP) - YEAR(NGSINH) AS Tuoi,

(SELECT AVG(YEAR(CURRENT\_TIMESTAMP) - YEAR(NGSINH)) FROM NHANVIEN) AS AvgAge

FROM NHANVIEN

GROUP BY MANV, HONV, TENLOT, TENNV, NGSINH;



47.Danh sách các Phòng ban, Tên đề án và tổng lương trung bình phải trả cho mỗi  
đề án (COMPUTE BY). Lương trung bình được tính từ trung bình của tổng  
lương\_đề án các nhân viên, trong đó lương\_đề án của mỗi nhân viên = LUONGx  
với số giờ làm việc cho mỗi đề án của nhân viên đó.

SELECT PHG.TENPHG, DA.TENDA, AVG(NV.LUONG \* PHANCONG.THOIGIAN) AS AvgSalary

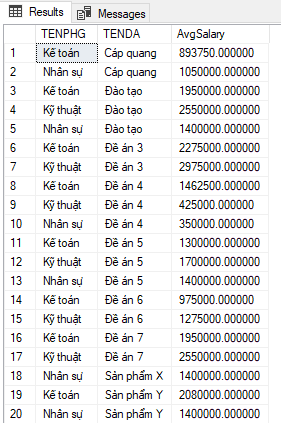
FROM PHONGBAN PHG

JOIN NHANVIEN NV ON PHG.MAPHG = NV.PHG

JOIN PHANCONG ON NV.MANV = PHANCONG.MA\_NVIEN

JOIN DEAN DA ON PHANCONG.SODA = DA.MADA

GROUP BY PHG.TENPHG, DA.TENDA;



**3.4 PHÉP CHIA**48.Danh sách những nhân viên MaNV, Phái, HoTen được nối từ HONV, TENLOT,  
TENNV làm việc trong mọi đề án của công ty

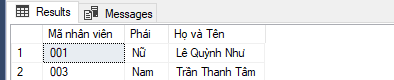
SELECT NHANVIEN.MANV 'Mã nhân viên', NHANVIEN.PHAI 'Phái', NHANVIEN.HONV + ' ' + NHANVIEN.TENLOT + ' ' + NHANVIEN.TENNV AS 'Họ và Tên'

FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MANV = PHANCONG.MA\_NVIEN

GROUP BY NHANVIEN.MANV, NHANVIEN.PHAI, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

HAVING COUNT(DISTINCT PHANCONG.SODA) < (SELECT COUNT(\*) FROM DEAN)

  
49.Danh sách những nhân viên MaNV, Phái, HoTen được nối từ HONV, TENLOT,  
TENNV được phân công tất cả đề án do phòng số 5 chủ trì.

SELECT NHANVIEN.MANV, NHANVIEN.PHAI, NHANVIEN.HONV + ' ' + NHANVIEN.TENLOT + ' ' + NHANVIEN.TENNV AS HoTen

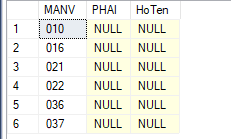
FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MANV = PHANCONG.MA\_NVIEN

WHERE PHANCONG.SODA IN (SELECT MADA FROM DEAN WHERE PHONG = 5)

GROUP BY NHANVIEN.MANV, NHANVIEN.PHAI, NHANVIEN.HONV, NHANVIEN.TENLOT, NHANVIEN.TENNV

HAVING COUNT(DISTINCT PHANCONG.SODA) < (SELECT COUNT(\*) FROM DEAN WHERE PHONG = 5)



50.Tìm những nhân viên MaNV, Phái, HoTen (HONV, TENLOT, TENNV) được phân  
công tất cả đề án mà nhân viên Lê Minh Tính làm việc

SELECT NV1.MANV, NV1.PHAI, NV1.HONV + ' ' + NV1.TENLOT + ' ' + NV1.TENNV AS HoTen

FROM NHANVIEN NV1

WHERE NOT EXISTS (

SELECT DA.MADA

FROM DEAN DA

WHERE EXISTS (

SELECT PC1.SODA

FROM PHANCONG PC1

WHERE PC1.MA\_NVIEN = 'Lê Minh Tính' AND PC1.SODA = DA.MADA

) AND NOT EXISTS (

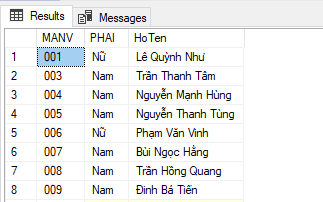
SELECT PC2.SODA

FROM PHANCONG PC2

WHERE PC2.MA\_NVIEN = NV1.MANV AND PC2.SODA = DA.MADA

)

)

  
51.Cho biết danh sách nhân viên tham gia vào tất cả các đề án ở TP HCM

SELECT NV.MANV, NV.PHAI, NV.HONV + ' ' + NV.TENLOT + ' ' + NV.TENNV AS HoTen

FROM NHANVIEN NV

WHERE NOT EXISTS (

SELECT DA.MADA

FROM DEAN DA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = 'TP.HCM' AND NOT EXISTS (

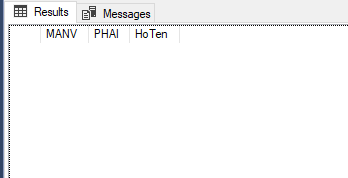
SELECT PC.SODA

FROM PHANCONG PC

WHERE PC.MA\_NVIEN = NV.MANV AND PC.SODA = DA.MADA

)

)

  
52.Cho biết phòng ban chủ trì tất cả các đề án ở HaNoi

SELECT PB.MAPHG, PB.TENPHG

FROM PHONGBAN PB

WHERE NOT EXISTS (

SELECT DA.MADA

FROM DEAN DA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = N'Hà Nội' AND NOT EXISTS (

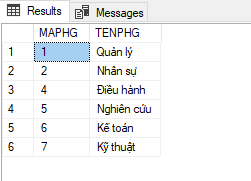
SELECT DA1.MADA

FROM DEAN DA1

WHERE DA1.PHONG = PB.MAPHG AND DA1.MADA = DA.MADA

)

)

  
**3.6. UNION/INTERSECT**53.Cho biết những phòng ban có nhân viên tham gia cả 2 dự án ở HaNoi và Tp.HCM.

SELECT PB.MAPHG, PB.TENPHG

FROM PHONGBAN PB

WHERE EXISTS (

SELECT NV.MANV

FROM NHANVIEN NV

INNER JOIN PHANCONG PC ON NV.MANV = PC.MA\_NVIEN

INNER JOIN DEAN DA ON PC.SODA = DA.MADA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = N'Hà Nội' AND NV.PHG = PB.MAPHG

)

INTERSECT

SELECT PB.MAPHG, PB.TENPHG

FROM PHONGBAN PB

WHERE EXISTS (

SELECT NV.MANV

FROM NHANVIEN NV

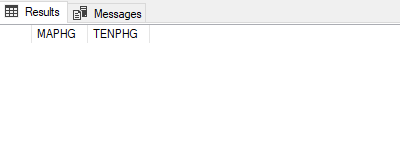
INNER JOIN PHANCONG PC ON NV.MANV = PC.MA\_NVIEN

INNER JOIN DEAN DA ON PC.SODA = DA.MADA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = N'TP.HCM' AND NV.PHG = PB.MAPHG

)

  
54. Cho biết những phòng ban có nhân viên tham gia dự án ở Tp.HCM hoặc ở HaNoi.

SELECT PB.MAPHG, PB.TENPHG

FROM PHONGBAN PB

WHERE EXISTS (

SELECT NV.MANV

FROM NHANVIEN NV

INNER JOIN PHANCONG PC ON NV.MANV = PC.MA\_NVIEN

INNER JOIN DEAN DA ON PC.SODA = DA.MADA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = N'TP.HCM' AND NV.PHG = PB.MAPHG

)

UNION

SELECT PB.MAPHG, PB.TENPHG

FROM PHONGBAN PB

WHERE EXISTS (

SELECT NV.MANV

FROM NHANVIEN NV

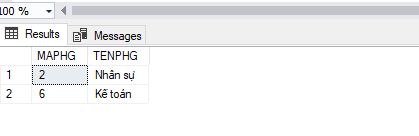
INNER JOIN PHANCONG PC ON NV.MANV = PC.MA\_NVIEN

INNER JOIN DEAN DA ON PC.SODA = DA.MADA

INNER JOIN DIADIEM\_PHG DD ON DA.PHONG = DD.MAPHG

WHERE DD.DIADIEM = N'HaNoi' AND NV.PHG = PB.MAPHG

)



**Module 4 & 5:**⮚ Nội dung: BATCH, VIEW, USER-DEFINED FUNCTION  
**PHẦN 1: BATCH**1. Viết đoạn batch để in ra tổng số dự án mà nhân viên có MANV = ‘333445555’  
tham gia. Nếu tổng số dự án <3 thì in ra “Tham gia ít” ngược lại in ra “ Tham  
gia nhiều”. Lưu ý, MaNV phải được sử dụng như biến, không gán giá trị trực  
tiếp vào điều kiện của câu truy vấn.

DECLARE @manv NVARCHAR(10);

DECLARE @count INT;

SET @manv = '333445555';

SELECT @count = COUNT(\*)

FROM PHANCONG

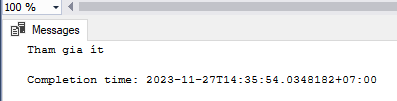
WHERE MA\_NVIEN = @manv;

IF @count < 3

PRINT 'Tham gia ít';

ELSE

PRINT 'Tham gia nhiều';

  
2. Viết vòng lặp while để xuất ra bảng cửu chương 2

DECLARE @i INT;

SET @i = 1;

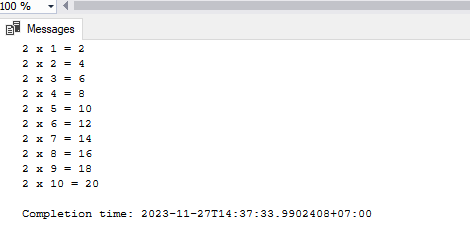
WHILE @i <= 10

BEGIN

PRINT CAST(2 AS NVARCHAR(10)) + ' x ' + CAST(@i AS NVARCHAR(10)) + ' = ' + CAST((2\*@i) AS NVARCHAR(10));

SET @i = @i + 1;

END

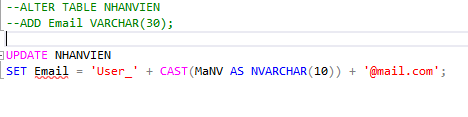
  
3. Thêm một cột Email(varchar(30)) vào bảng NHANVIEN. Viết vòng lặp WHILE để chèn dữ liệu cho cột Email cho các nhân viên theo mẫu sau :  
User\_MaNV@mail.com.  
Ví dụ, nhân viên có mã số 101 được thêm vào dữ liệu : [User\_101@mail.com](mailto:User_101@mail.com)

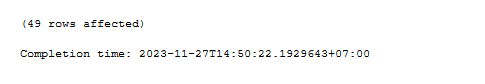
ALTER TABLE NHANVIEN

ADD Email VARCHAR(30);

UPDATE NHANVIEN

SET Email = 'User\_' + CAST(MaNV AS NVARCHAR(10)) + '@mail.com';





  
4. Viết đoạn batch sử dụng SELECT …CASE để trả về thông tin của nhân viên  
cùng với độ tuổi của nhân viên. Trong đó, độ tuổi tính như sau :nếu tuổi của  
nhân viên từ 18🡪30 là độ tuổi thanh niên, từ 31 đến 45 là trung niên, từ 46  
đến 60 là cao niên.

SELECT

MaNV,

HONV

NGSINH,

CASE

WHEN DATEDIFF(YEAR, NGSINH, GETDATE()) BETWEEN 18 AND 30 THEN 'Thanh niên'

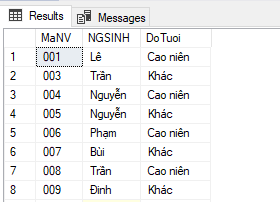
WHEN DATEDIFF(YEAR, NGSINH, GETDATE()) BETWEEN 31 AND 45 THEN 'Trung niên'

WHEN DATEDIFF(YEAR, NGSINH, GETDATE()) BETWEEN 46 AND 60 THEN 'Cao niên'

ELSE 'Khác'

END AS DoTuoi

FROM NHANVIEN;



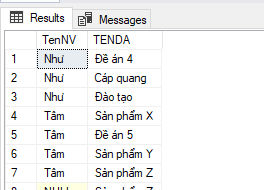
**Phần 2 : VIEWS**1. Viết truy vấn liệt kê tên các nhân viên cùng với các dự án mà nhân viên đó  
tham gia. Run và kiểm tra kết quả.

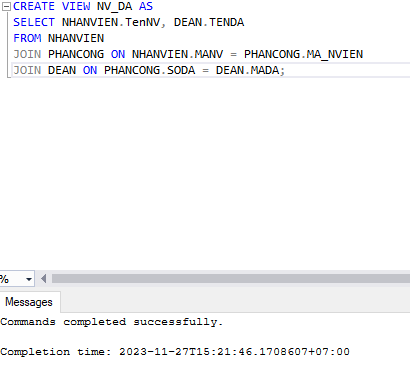
SELECT NHANVIEN.TenNV, DEAN.TENDA

FROM NHANVIEN

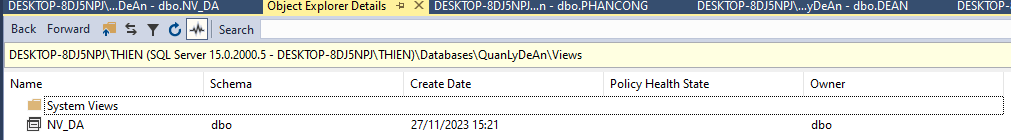
Join PHANCONG on NHANVIEN.MANV = PHANCONG.MA\_NVIEN

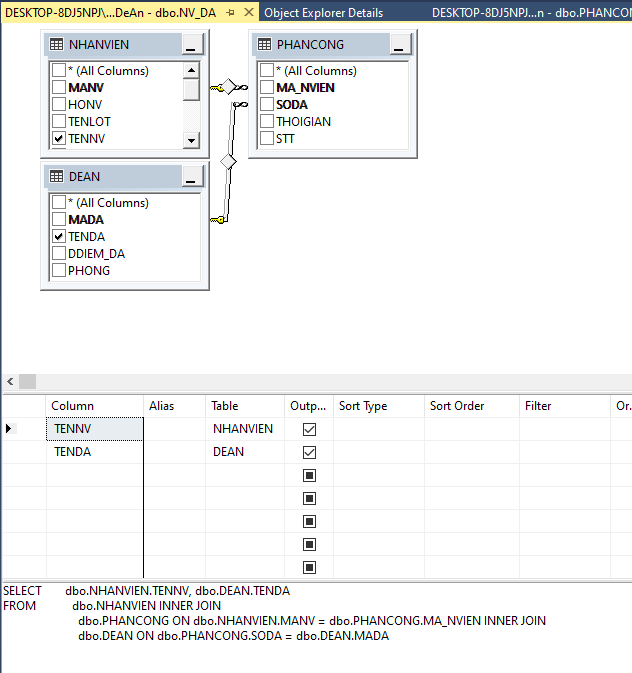
JOIN DEAN ON PHANCONG.SODA = DEAN.MADA

  
2. Tạo một View có tên NV\_DA với nội dung truy vấn là câu 1. Mở cửa sổ Object  
Explorer, xem view vừa tạo lưu ở đâu?. Xem kết quả dữ liệu từ View vừa tạo  
(SELECT \* FROM NV\_DA) và so sánh với kết quả ở câu 1; có khác nhau  
không?. Tại sao?



Kết quả của câu truy vấn từ View NV\_DA sẽ giống hệt với kết quả của câu truy vấn gốc, vì View NV\_DA được tạo dựa trên câu truy vấn đó.





**3.** Thêm một dòng tùy ý vào bảng PHANCONG, sau đó thực hiện lệnh SELECT \*  
FROM NV\_DA, kết quả có thay đổi so với câu 2 không?. Tại sao?

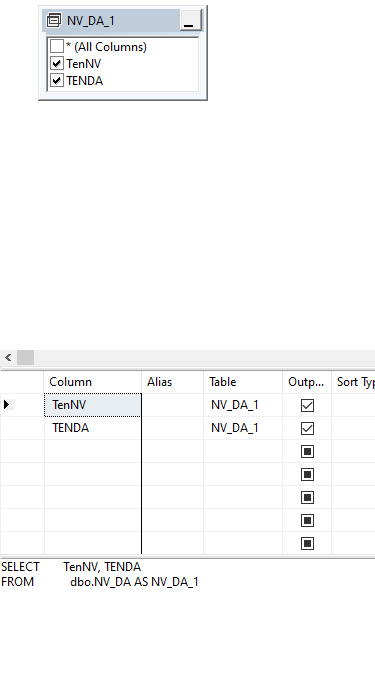
Khác vì dữ liệu được truy vấn từ View NV\_DA cũng sẽ thay đổi tương ứng theo các bảng PHANCONG

ALTER TABLE PHANCONG

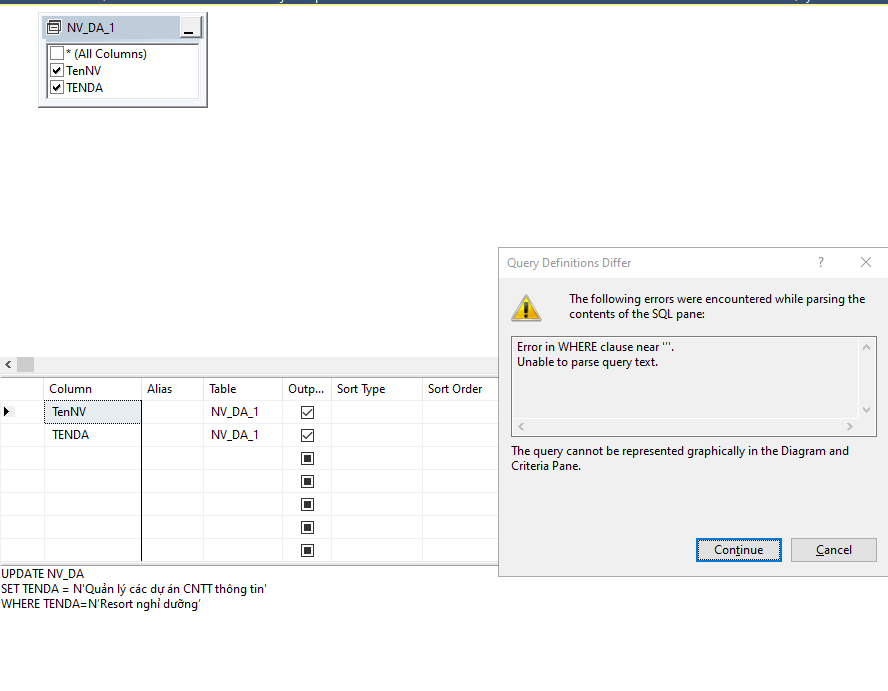
NOCHECK CONSTRAINT ALL

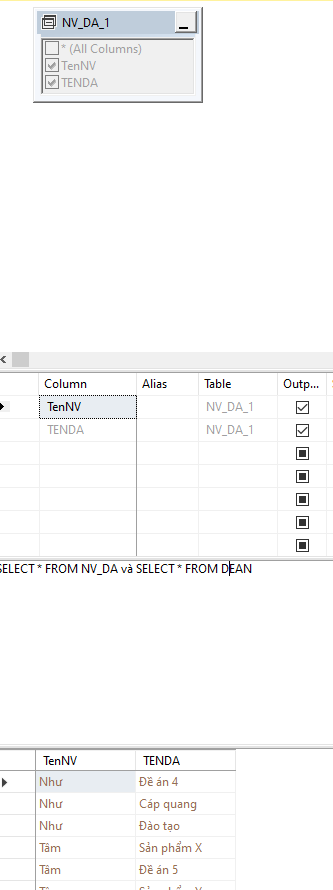
Insert into PHANCONG(MA\_NVIEN,SODA,THOIGIAN,STT)

Values('002',10,5,2)

  
**4.** Thực hiện lệnh:  
UPDATE NV\_DA  
SET TENDA = N‘ Quản lý các dự án CNTT thông tin’

WHERE TENDA=N’Resort nghỉ dưỡng’  
Câu lệnh có thực hiện được không?. Tại sao?

  
Thực hiện câu lệnh: SELECT \* FROM NV\_DA và SELECT \* FROM DEAN để xem  
sự thay đổi dữ liệu

  
5. Viết truy vấn hiển thị HoTen của nhân viên, tên dự án nhân viên đó tham gia  
cùng với tổng tiền lương của nhân viên theo dự án (tiền lương = số giờ \*  
LUONG). Sắp xếp tiền lương tăng dần. Sau đó, thực hiện các yêu cầu sau:

SELECT NHANVIEN.HONV, DEAN.TENDA, SUM(PHANCONG.THOIGIAN \* NHANVIEN.LUONG) AS 'TongLuong'

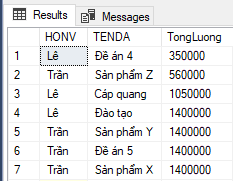
FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MaNV = PHANCONG.MA\_NVIEN

JOIN DEAN ON PHANCONG.SODA = DEAN.MADA

GROUP BY NHANVIEN.HONV, DEAN.TENDA

ORDER BY 'TongLuong' ASC;

  
a. Tạo một View5a dựa trên query này. Bạn sẽ gặp lỗi, tại sao?

CREATE VIEW View5a AS

SELECT NHANVIEN.HONV, DEAN.TENDA, SUM(PHANCONG.THOIGIAN \* NHANVIEN.LUONG) AS 'TongLuong'

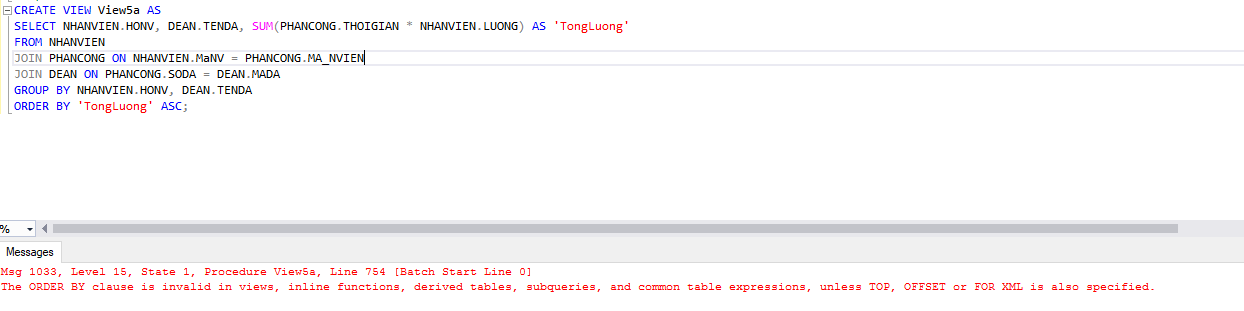
FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MaNV = PHANCONG.MA\_NVIEN

JOIN DEAN ON PHANCONG.SODA = DEAN.MADA

GROUP BY NHANVIEN.HONV, DEAN.TENDA

ORDER BY 'TongLuong' ASC;



The ORDER BY clause is invalid in views, inline functions, derived tables, subqueries, and common table expressions, unless TOP, OFFSET or FOR XML is also specified.

Mệnh đề ORDER BY không hợp lệ trong các dạng xem, hàm nội tuyến, bảng dẫn xuất, truy vấn con và biểu thức bảng chung, trừ khi TOP, OFFSET hoặc FOR XML cũng được chỉ định.

b. Hãy hiệu chỉnh để có thể tạo được View5b dựa trên query trên.

CREATE VIEW View5b AS

SELECT

NHANVIEN.HONV,

DEAN.TENDA,

SUM(PHANCONG.THOIGIAN \* NHANVIEN.LUONG) AS TongLuong

FROM

NHANVIEN

JOIN

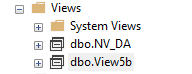
PHANCONG ON NHANVIEN.MaNV = PHANCONG.MA\_NVIEN

JOIN

DEAN ON PHANCONG.SODA = DEAN.MADA

GROUP BY

NHANVIEN.HONV, DEAN.TENDA;



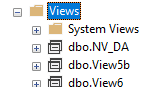
6. Tạo View6 gồm các thông tin của bảng DEAN nhưng chỉ lấy các dự án có địa  
điểm ở Tp.HCM.

CREATE VIEW View6 AS

SELECT \*

FROM DEAN

WHERE DDIEM\_DA = 'TP.HCM';



a. Sửa câu View6 thành View6a có thêm tùy chọn WITH ENCRYPTION.  
Sử dụng sp\_helptext để xem lại mã lệnh. Có xem được không?

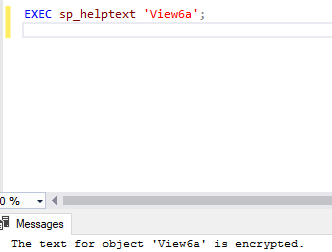
CREATE VIEW View6a WITH ENCRYPTION AS

SELECT \*

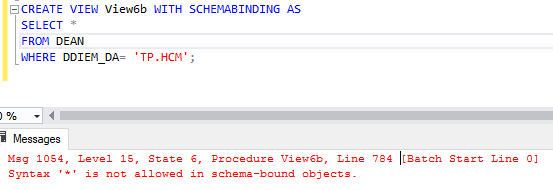
FROM DEAN

WHERE DDIEM\_DA = 'TP.HCM';

 mã nguồn của View sẽ được mã hóa và sẽ không thể xem lại nó bằng cách sử dụng sp\_helptext

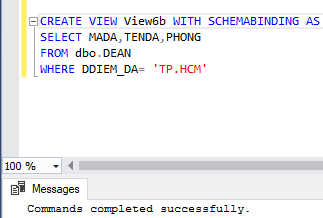


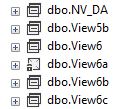
b. Sửa câu View6 thành View6b có thêm tùy chọn WITH  
SCHEMABINDING. Bạn cần hiệu chỉnh điều gì để có thể thực thi được  
view.



Sử dụng tùy chọn WITH SCHEMABINDING, cần đảm bảo rằng tất cả các đối tượng được tham chiếu trong View (trong trường hợp này là bảng DEAN và cột DDIEM\_DA) phải tồn tại. Nếu không sẽ nhận được một lỗi

Cần phải bao gồm tên schema trước tên bảng và cột : dbo.DEAN và phải chọn từng column một

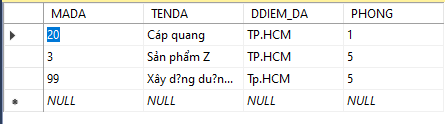
  
c. Sửa câu View6 thành View6c có thêm tùy chọn WITH CHECK OPTION.  
Sau đó thêm một dòng dữ liệu sau thông qua View6c:  
INSERT View6c  
VALUES(‘99’,’Xây dựng đường cao tốc LT-DN’, ‘Đồng Nai’,’5’)

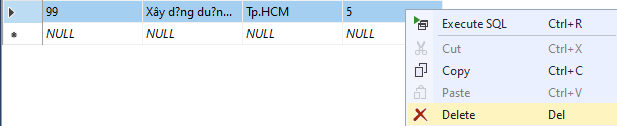
  
Bạn có thêm được không, tại sao?

Không thêm được vì dữ liệu đó tuân thủ các ràng buộc của View6c. Trong trường hợp này, dữ liệu phải tuân thủ ràng buộc DDIEM\_DA = 'TP.HCM' và ở đây là “Đồng Nai’ 🡪 không thêm được  
Sửa lại dữ liệu để có thể thêm được thông qua View6c.

INSERT INTO View6c

VALUES('99','Xây dựng đường cao tốc LT-DN', 'Tp.HCM','5');

  
d. Xóa dữ liệu vừa thêm ở câu c thông qua View6c.

  
7. Tạo 3 bảng DEAN\_HCM, DEAN\_HANOI, DEAN\_VT gồm các cột:

MaDA varchar(2) Primary Key, TenDA nvarchar(50), DDIEM\_DA

varchar(20), PHONG varchar(2). Trong đó :

- DEAN\_HCM: check contraint của DDIEM\_DA là ‘Tp.Hồ Chí Minh’

- DEAN\_HANOI: check contraint của DDIEM\_DA là ‘Hà Nội’

- DEAN\_VT: check contraint của DDIEM\_DA là ‘Vũng Tàu’

CREATE TABLE DEAN\_HCM (

MaDA varchar(2) PRIMARY KEY,

TenDA nvarchar(50),

DDIEM\_DA varchar(20) CHECK (DDIEM\_DA = 'Tp.Hồ Chí Minh'),

PHONG varchar(2)

);

CREATE TABLE DEAN\_HANOI (

MaDA varchar(2) PRIMARY KEY,

TenDA nvarchar(50),

DDIEM\_DA varchar(20) CHECK (DDIEM\_DA = 'Hà Nội'),

PHONG varchar(2)

);

CREATE TABLE DEAN\_VT (

MaDA varchar(2) PRIMARY KEY,

TenDA nvarchar(50),

DDIEM\_DA varchar(20) CHECK (DDIEM\_DA = 'Vũng Tàu'),

PHONG varchar(2)

);



Thêm vào mỗi bảng 2 record với ràng buộc tương ứng.

INSERT INTO DEAN\_HCM VALUES ('01', 'Dự án 1', 'Tp.Hồ Chí Minh', 'A1');

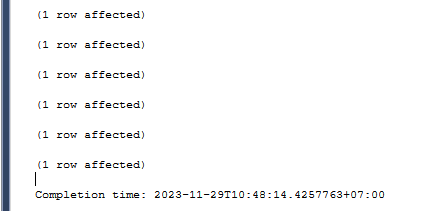
INSERT INTO DEAN\_HCM VALUES ('02', 'Dự án 2', 'Tp.Hồ Chí Minh', 'A2');

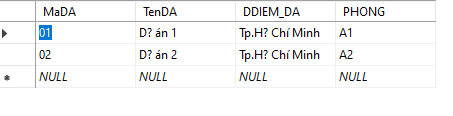
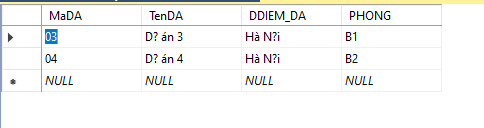
INSERT INTO DEAN\_HANOI VALUES ('03', 'Dự án 3', 'Hà Nội', 'B1');

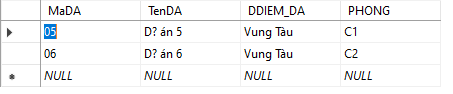
INSERT INTO DEAN\_HANOI VALUES ('04', 'Dự án 4', 'Hà Nội', 'B2');

INSERT INTO DEAN\_VT VALUES ('05', 'Dự án 5', 'Vũng Tàu', 'C1');

INSERT INTO DEAN\_VT VALUES ('06', 'Dự án 6', 'Vũng Tàu', 'C2');





Tạo một partition view từ 3 bảng dữ liệu trên.

CREATE VIEW DEAN\_VIEW AS

SELECT \* FROM DEAN\_HCM

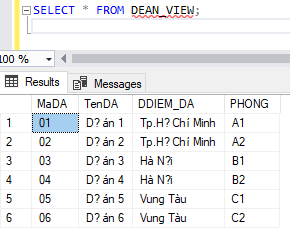
UNION ALL

SELECT \* FROM DEAN\_HANOI

UNION ALL

SELECT \* FROM DEAN\_VT;

Xem kết quả view vừa tạo được.



**PHẦN 3: USER-DEFINED FUNCTIONS**

1. Viết hàm trả về tổng tiền lương trung bình của một phòng ban tùy ý (truyền  
vào MaPB(MAPHG)

CREATE FUNCTION fn\_avg\_salary(@MAPHG varchar(10))

RETURNS float

AS

BEGIN

DECLARE @avg\_salary float;

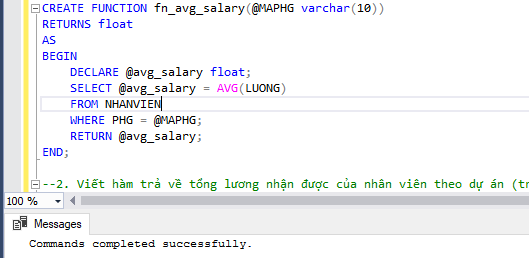
SELECT @avg\_salary = AVG(LUONG)

FROM NHANVIEN

WHERE PHG = @MAPHG;

RETURN @avg\_salary;

END;

  
2. Viết hàm trả về tổng lương nhận được của nhân viên theo dự án (truyền vào  
MANV và MADA)

Vì hàm fn\_avg\_salary đã tồn tại , phải xóa nó trước khi tạo hàm mới

IF OBJECT\_ID('fn\_avg\_salary', 'FN') IS NOT NULL

DROP FUNCTION fn\_avg\_salary;

Rồi mới tạo :

CREATE FUNCTION fn\_total\_salary(@MANV varchar(10), @MADA varchar(10))

RETURNS float

AS

BEGIN

DECLARE @total\_salary float;

SELECT @total\_salary = SUM(PHANCONG.THOIGIAN \* NHANVIEN.LUONG)

FROM NHANVIEN

JOIN PHANCONG ON NHANVIEN.MaNV = PHANCONG.MA\_NVIEN

WHERE NHANVIEN.MaNV = @MANV AND PHANCONG.MA\_NVIEN = @MADA;

RETURN @total\_salary;

END;  
3. Viết hàm trả về tổng tiền lương trung bình của các phòng ban

CREATE FUNCTION fn\_avg\_salary\_departments()

RETURNS TABLE

AS

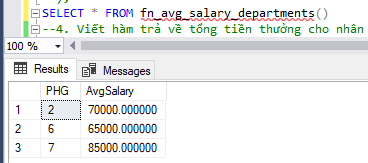
RETURN (

SELECT PHG, AVG(LUONG) as AvgSalary

FROM NHANVIEN

GROUP BY PHG

);

  
4. Viết hàm trả về tổng tiền thưởng cho nhân viên dựa vào tổng số giờ tham gia  
dự án(Time\_Total) như sau:  
- Nếu Time\_Total >=30 và <=60 thì tổng tiền thưởng = 500 ($)  
- Nếu Time\_Total >60 và <100 thì tổng tiền thưởng = 1000 ($)  
- Nếu Time\_Total >=100 và <150 thì tổng tiền thưởng =1200($)  
- Nếu Time\_Total >=150 thì tổng tiền thưởng = 1600 ($)

CREATE FUNCTION fn\_bonus(@Time\_Total int)

RETURNS float

AS

BEGIN

DECLARE @bonus float;

IF @Time\_Total BETWEEN 30 AND 60

SET @bonus = 500;

ELSE IF @Time\_Total > 60 AND @Time\_Total < 100

SET @bonus = 1000;

ELSE IF @Time\_Total >= 100 AND @Time\_Total < 150

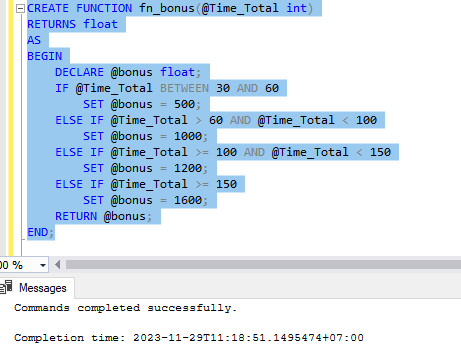
SET @bonus = 1200;

ELSE IF @Time\_Total >= 150

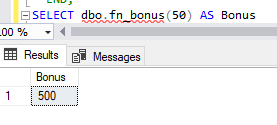
SET @bonus = 1600;

RETURN @bonus;

END;



Truyền vào 50 thử :

  
5. Viết hàm trả ra tổng số dự án theo mỗi phòng ban.

CREATE FUNCTION GetProjectCountByDepartment()

RETURNS TABLE

AS

RETURN

(

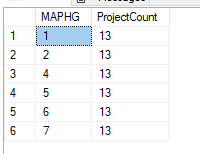
SELECT MAPHG, COUNT(MADA) as ProjectCount

FROM DEAN , PHONGBAN

GROUP BY MAPHG

)

Check : SELECT \* FROM GetProjectCountByDepartment()



6. Viết hàm trả về kết quả là một bảng (Table), viết bằng  
hai cách: Inline Table-Valued Functions và  
Multistatement Table-Valued. Thông tin gồm: MaNV,  
HoTen, NgaySinh, NguoiThan, TongLuongTB(MANV,HONV,TENNV,NGSINH,TENTN)

Inline Table-Valued Functions

CREATE FUNCTION GetEmployeeDetailsInline()

RETURNS TABLE

AS

RETURN

(

SELECT NHANVIEN.MANV, NHANVIEN.HONV, NHANVIEN.TENNV, NHANVIEN.NGSINH, THANNHAN.TENTN

FROM NHANVIEN

INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

INNER JOIN THANNHAN ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

)

Multistatement Table-Valued

CREATE FUNCTION GetEmployeeDetailsMultistatement()

RETURNS @EmployeeDetailsTable TABLE

(

MANV NVARCHAR(10),

HONV NVARCHAR(50),

TENNV NVARCHAR(50),

NGSINH DATE,

TENTN NVARCHAR(50)

)

AS

BEGIN

INSERT INTO @EmployeeDetailsTable

SELECT NHANVIEN.MANV, NHANVIEN.HONV, NHANVIEN.TENNV, NHANVIEN.NGSINH, THANNHAN.TENTN

FROM NHANVIEN

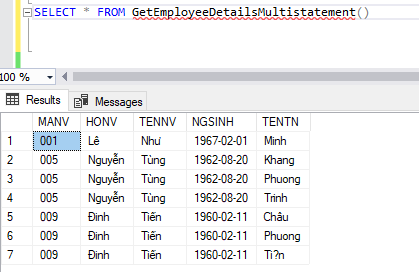
INNER JOIN PHONGBAN ON NHANVIEN.PHG = PHONGBAN.MAPHG

INNER JOIN THANNHAN ON NHANVIEN.MANV = THANNHAN.MA\_NVIEN

RETURN

END

Check lại bằng : SELECT \* FROM GetEmployeeDetailsMultistatement()



**Module 7 & 8:**⮚ Nội dung: Stored Procedure, Cursor  
1. Việc insert, update, delete, select dữ liệu ở bảng hoàn toàn thực hiện được  
chỉ bằng câu lệnh DML. Tuy nhiên, trong một số trường hợp, chúng ta cần  
thiết phải gán quyền cho người dùng (sẽ được đề cập ở module sau) mà  
không cần thực hiện việc gán quyền trên table thì ta sẽ sử dụng procedure để  
điều khiển thao tác dữ liệu. Ở bài tập này, yêu cầu sinh viên tạo thủ tục với  
các thao tác tương ứng như sau:  
a. Tạo thủ tục IN\_PROC\_DEAN cho phép thêm dữ liệu vào bảng đề án với  
các tham số vào là các field của bảng DEAN

CREATE PROCEDURE IN\_PROC\_DEAN

@MADA VARCHAR(2),

@TENDA NVARCHAR(50),

@DDIEM\_DA VARCHAR(20),

@PHONG NVARCHAR(2)

AS

BEGIN

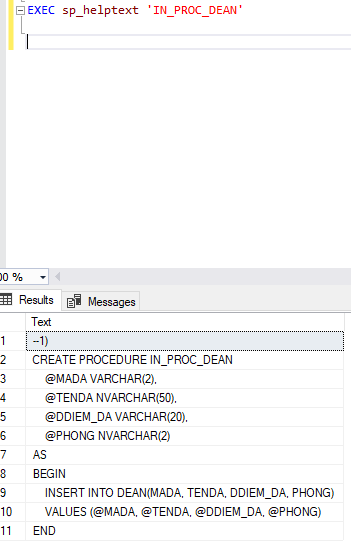
INSERT INTO DEAN(MADA, TENDA, DDIEM\_DA, PHONG)

VALUES (@MADA, @TENDA, @DDIEM\_DA, @PHONG)

END

Dùng lệnh để kiểm tra sự tồn tại của thủ tục : EXEC sp\_helptext 'IN\_PROC\_DEAN'

Thêm dữ liệu vào : EXEC IN\_PROC\_DEAN @MADA = '50', @TENDA = N'Trồng cây siêu tăng trưởng', @DDIEM\_DA = N'Vũng Tàu', @PHONG = '5'



  
b. Tạo thủ tục SE\_PRO\_DEAN cho phép xem các đề án có địa điểm đề án  
được truyền vào khi gọi thủ tục.

CREATE PROCEDURE SE\_PRO\_DEAN

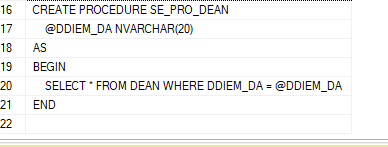
@DDIEM\_DA NVARCHAR(20)

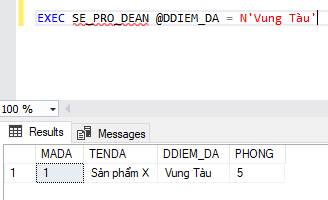
AS

BEGIN

SELECT \* FROM DEAN WHERE DDIEM\_DA = @DDIEM\_DA

END



  
c. Tạo thủ tục UP\_PROC\_DEAN cho phép cập nhật lại địa điểm đề án là Bà  
Rịa Vũng Tàu cho các đề án có địa điểm Vung Tau với 2 tham số truyền  
vào là diadiem\_old, diadiem\_new.

CREATE PROCEDURE UP\_PROC\_DEAN

@diadiem\_old VARCHAR(20),

@diadiem\_new VARCHAR(20)

AS

BEGIN

UPDATE DEAN

SET DDIEM\_DA = @diadiem\_new

WHERE DDIEM\_DA = @diadiem\_old;

END;

EXEC UP\_PROC\_DEAN 'Vung Tàu', 'Bà Rịa Vũng Tàu';

  
d. Tạo thủ tục DEL\_PROC\_DEAN cho phép xóa các đề án đã thêm vào  
thông qua thủ tục IN\_PROC\_DEAN. Tham số truyền vào là MaDean.

CREATE PROCEDURE DEL\_PROC\_DEAN

@MaDean VARCHAR(2)

AS

BEGIN

DELETE FROM DEAN

WHERE MADA = @MaDean;

END;

Xóa đề án 50 : EXEC DEL\_PROC\_DEAN '50';

  
2. Tạo thủ tục có tham số vào là MaNV, tham số ra (OUTPUT) là tổng số giờ  
(tổng thời gian) làm việc của nhân viên của tất cả các dự án.  
CREATE PROC proc\_name  
(@manv datatype, @tong datatype OUTPUT)  
AS  
BEGIN  
--SELECT statement  
END  
--exec  
DECLARE @tong datatype  
EXEC proc\_name ‘Manv\_data’, @tong OUTPUT

CREATE PROCEDURE TongGioLam

@MaNV NVARCHAR(50),

@tonggio INT OUTPUT

AS

BEGIN

SELECT @tonggio = SUM(THOIGIAN)

FROM PHANCONG

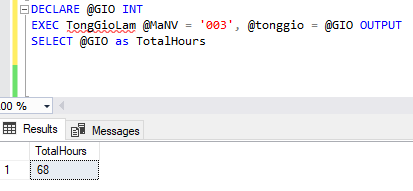
WHERE MA\_NVIEN = @MaNV

END

DECLARE @GIO INT

EXEC TongGioLam @MaNV = '003', @tonggio = @GIO OUTPUT

SELECT @GIO as TotalHours



3. Viết thủ tục có tên INSERT\_DEAN để thêm dữ liệu vào bảng DEAN thỏa mãn  
các ràng buộc sau: kiểm tra MADEAN cần chèn có rỗng không hoặc có trùng  
với các mã đề án khác đã có trong bảng đề án không. Nếu có hãy thông báo lỗi  
“ Bị trùng mã đề án hoặc mã đề án rỗng, chọn mã đề án khác!!!”.

CREATE PROCEDURE INSERT\_DEAN

@TENDA NVARCHAR(50),

@MADA varchar(2),

@DDIEM\_DA NVARCHAR(20),

@PHONG varchar(2)

AS

BEGIN

IF @MADA IS NULL OR EXISTS (SELECT 1 FROM DEAN WHERE MADA = @MADA)

BEGIN

print N'Bị trùng mã đề án hoặc mã đề án rỗng, chọn mã đề án khác!!!';

return

END

ELSE

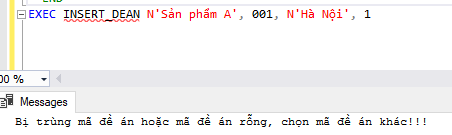
BEGIN

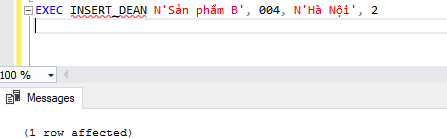
INSERT INTO DEAN (TENDA, MADA, DDIEM\_DA, PHONG)

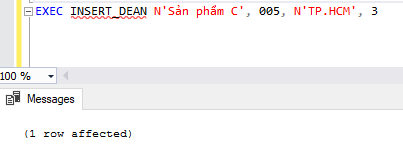
VALUES (@TENDA, @MADA, @DDIEM\_DA, @PHONG)

END

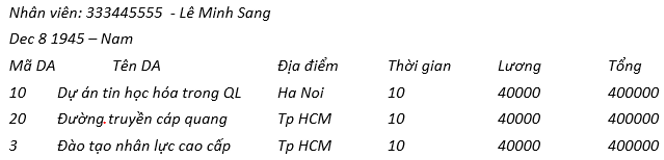
Thực thi thủ tục với 1 trường hợp đúng và 2 trường hợp sai để kiểm chứng.







  
4. Viết thủ tục SALARY\_REPORT có sử dụng con trỏ (cursor) để in ra kết quả  
cửa sổ query analyzer là thông tin giờ dự án cùng với tổng tiền theo dự án  
của từng nhân viên.(Xem mẫu ví dụ dưới cho 1 nhân viên)



CREATE PROCEDURE SALARY\_REPORT

AS

BEGIN

DECLARE @MANV VARCHAR(9), @HONV NVARCHAR(15), @TENLOT NVARCHAR(30), @TENNV NVARCHAR(30), @NGSINH SMALLDATETIME, @PHAI NVARCHAR(3), @LUONG NUMERIC(18,0)

DECLARE @MADA VARCHAR(2), @TENDA NVARCHAR(50), @DDIEM\_DA VARCHAR(20), @PHONG VARCHAR(2), @THOIGIAN INT

DECLARE @Total MONEY

DECLARE employee\_cursor CURSOR FOR

SELECT MANV, HONV, TENLOT, TENNV, NGSINH, PHAI, LUONG

FROM NHANVIEN

OPEN employee\_cursor

FETCH NEXT FROM employee\_cursor INTO @MANV, @HONV, @TENLOT, @TENNV, @NGSINH, @PHAI, @LUONG

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'Nhân viên: ' + @MANV + ' – ' + @HONV + ' ' + @TENLOT + ' ' + @TENNV

PRINT CONVERT(VARCHAR, @NGSINH, 107) + ' - ' + @PHAI

PRINT 'Mã DA Tên DA Địa điểm Thời gian Lương Tổng'

DECLARE project\_cursor CURSOR FOR

SELECT MADA, TENDA, DDIEM\_DA, PHONG, THOIGIAN

FROM PHANCONG JOIN DEAN ON PHANCONG.SODA = DEAN.MADA

WHERE MA\_NVIEN = @MANV

OPEN project\_cursor

FETCH NEXT FROM project\_cursor INTO @MADA, @TENDA, @DDIEM\_DA, @PHONG, @THOIGIAN

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @Total = @THOIGIAN \* @LUONG

PRINT @MADA + ' ' + @TENDA + ' ' + @DDIEM\_DA + ' ' + CONVERT(VARCHAR, @THOIGIAN) + ' ' + CONVERT(VARCHAR, @LUONG) + ' ' + CONVERT(VARCHAR, @Total)

FETCH NEXT FROM project\_cursor INTO @MADA, @TENDA, @DDIEM\_DA, @PHONG, @THOIGIAN

END

CLOSE project\_cursor

DEALLOCATE project\_cursor

FETCH NEXT FROM employee\_cursor INTO @MANV, @HONV, @TENLOT, @TENNV, @NGSINH, @PHAI, @LUONG

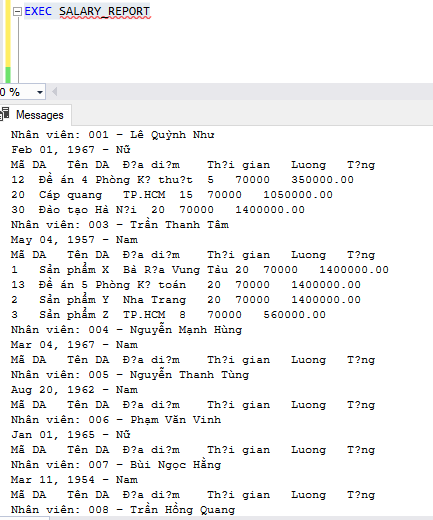
END

CLOSE employee\_cursor

DEALLOCATE employee\_cursor

END

EXEC SALARY\_REPORT



5. Viết thủ tục SUM\_SALARY để tính tổng tiền phải trả cho nhân viên với tham  
số truyền vào là mã nhân viên.  
Ví dụ: Tổng tiền phải trả cho nhân viên ‘333445555’ là  
400000+400000+400000=1200000.

CREATE PROCEDURE SUM\_SALARY

@MANV NVARCHAR(50),

@TotalSalary INT OUTPUT

AS

BEGIN

SELECT @TotalSalary = SUM(LUONG)

FROM NHANVIEN

WHERE MANV = @MANV

END

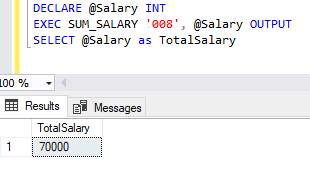
**Check lại với :**

**Khai báo biến :**

DECLARE @Salary INT

EXEC SUM\_SALARY '008', @Salary OUTPUT

SELECT @Salary as TotalSalary



6. Viết thủ tục để kiểm tra việc thêm dữ liệu vào bảng PHANCONG thỏa mãn  
yêu cầu sau: THOIGIAN phải > 0 và SoDA phải tồn tại ở bảng DeAn. Nếu  
không thỏa mãn phải thông báo lỗi tương ứng và không được phép insert.  
--*hướng dẫn  
CREATE PROC proc\_name  
AS  
BEGIN TRY  
--do something  
END TRY  
BEGIN CATCH  
ROLLBACK TRANSACTION  
PRINT ‘Message content’  
END CATCH*

CREATE PROCEDURE INSERT\_PHANCONG

@MA\_NVIEN NVARCHAR(9),

@SODA INT,

@STT INT,

@THOIGIAN INT

AS

BEGIN

IF @THOIGIAN <= 0

BEGIN

PRINT N'THOIGIAN phải lớn hơn 0';

RETURN;

END

IF NOT EXISTS (SELECT 1 FROM DEAN WHERE MADA = @SODA)

BEGIN

PRINT N'SoDA phải tồn tại trong bảng DeAn';

RETURN;

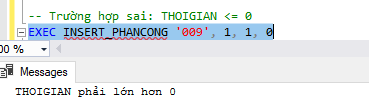
END

INSERT INTO PHANCONG (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES (@MA\_NVIEN, @SODA, @STT, @THOIGIAN)

END

**

**7. Thêm vào cột HIRE\_DATE (ngày nghỉ hưu) trong bảng nhân viên. Viết thủ tục  
dùng để cập nhật ngày nghỉ hưu là ngày hiện tại cộng thêm 100 (ngày) cho những nhân viên có tuổi từ 65 trở lên. Tham số truyền vào là field ngaysinh  
của nhân viên.

--Thêm cột HIRE\_DAY

ALTER TABLE NHANVIEN

ADD HIRE\_DATE DATE;

CREATE PROCEDURE UPDATE\_HIRE\_DATE

@NGSINH DATE

AS

BEGIN

DECLARE @currentAge INT;

SET @currentAge = DATEDIFF(YEAR, @NGSINH, GETDATE());

IF @currentAge >= 65

BEGIN

UPDATE NHANVIEN

SET HIRE\_DATE = DATEADD(DAY, 100, GETDATE())

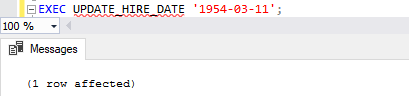
WHERE NGSINH = @NGSINH;

END

END

Test thử :

EXEC UPDATE\_HIRE\_DATE '1954-03-11';





**Module 9 & 10:**⮚ Nội dung: Triggers, Users role  
1. Tạo triggers trên bảng nhân viên cho thao tác UPDATE. Khi có thao tác UPDATE xảy ra trên field TENNV thì thông báo cho người dùng ‘Không được cập nhật’ và hủy thao tác.

CREATE TRIGGER trg\_Update\_TENNV

ON NHANVIEN

FOR UPDATE

AS

BEGIN

IF UPDATE(TENNV)

BEGIN

THROW 51000, N'Không được cập nhật', 1;

ROLLBACK TRANSACTION;

END

END

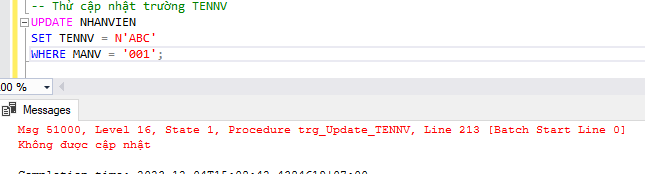
Check lại

Thử cập nhật trường TENNV :

UPDATE NHANVIEN

SET TENNV = N'ABC'

WHERE MANV = '001';

  
2. Thêm một cột ToTal\_Time vào trong bảng nhân viên. Viết trigger cho thao tác insert,update, delete trên bảng PhanCong. Khi có mẫu tin được thêm vào, hay cập nhật, hay xóa thì ToTal\_Time được tính lại tương ứng cho nhân viên được phân công. Lưu ý, giả sử ban đầu ToTal\_Time = 0. ToTal\_Time là tổng thời gian phân công cho nhân viên của các dự án.

thêm cột ToTal\_Time vào bảng NHANVIEN

ALTER TABLE NHANVIEN

ADD ToTal\_Time INT DEFAULT 0;

CREATE TRIGGER trg\_PhanCong

ON PhanCong

AFTER INSERT, UPDATE, DELETE

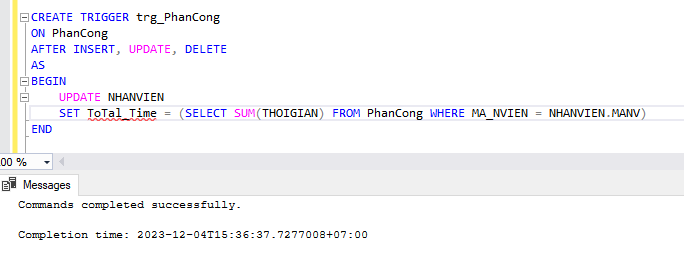
AS

BEGIN

UPDATE NHANVIEN

SET ToTal\_Time = (SELECT SUM(THOIGIAN) FROM PhanCong WHERE MA\_NVIEN = NHANVIEN.MANV)

END



Thêm một bản ghi vào bảng PhanCong

INSERT INTO PhanCong (MA\_NVIEN, SODA, STT, THOIGIAN)

VALUES ('010', 1, 1, 32)

-- Cập nhật một bản ghi trong bảng PhanCong

UPDATE PhanCong

SET THOIGIAN = 40

WHERE MA\_NVIEN = '010' AND SODA = 1

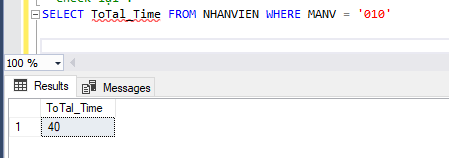
-- Xóa một bản ghi từ bảng PhanCong

DELETE FROM PhanCong

WHERE MA\_NVIEN = '010' AND SODA = 1

Check lại :

SELECT ToTal\_Time FROM NHANVIEN WHERE MANV = '010'



3. Tạo trigger cho thao tác insert, update trên bảng Nhanvien để kiểm tra ràng buộc liên thuộc tính giữa ngày sinh và Hire\_date, trong đó ngày sinh luôn luôn < Hire\_date – 40. (40 là giá trị của datepart ‘ yy’).

CREATE TRIGGER trg\_Nhanvien

ON Nhanvien

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted i WHERE i.NGSINH >= DATEADD(yy, -40, i.HIRE\_DATE))

BEGIN

RAISERROR ('ngaysinh phải nhỏ hơn Hire\_date - 40 năm', 16, 1);

ROLLBACK TRANSACTION;

END

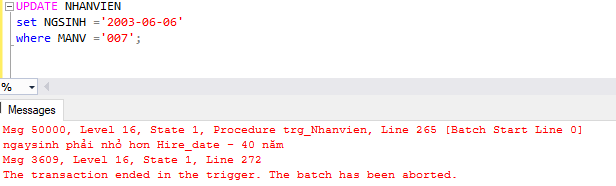
END

--Test :

UPDATE NHANVIEN

set NGSINH ='2003-06-06'

where MANV ='007'



4. Tạo trigger để kiểm tra thao tác insert trên bảng THANNHAN sao cho số lượng thân nhân được thêm vào tương ứng với một nhân viên không quá 5 người.

CREATE TRIGGER trg\_THANNHAN

ON THANNHAN

AFTER INSERT

AS

BEGIN

DECLARE @NhanvienID INT;

SELECT @NhanvienID = MA\_NVIEN FROM inserted;

IF (SELECT COUNT(\*) FROM THANNHAN WHERE MA\_NVIEN = @NhanvienID) > 5

BEGIN

RAISERROR ('Mỗi nhân viên không được có quá 5 người thân', 16, 1);

ROLLBACK TRANSACTION;

END

END;

--Check thử :

INSERT INTO THANNHAN (MA\_NVIEN,TENTN)

VALUES ('005', 'A'),

('005', 'B'),

('005', 'C'),

('005', 'D'),

('005', 'E');

