

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM
THÀNH PHỐ HỒ CHÍ MINH



Bộ môn Hệ thống thông tin – Khoa Công nghệ thông tin

SÁCH HƯỚNG DẪN THỰC HÀNH
CƠ SỞ DỮ LIỆU
(Lưu hành nội bộ)

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2022

Bộ môn Hệ thống thông tin – Khoa Công nghệ thông tin

**SÁCH HƯỚNG DẪN THỰC HÀNH
CƠ SỞ DỮ LIỆU**

(Tài liệu dùng cho hệ Đại học)

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2012

LỜI NÓI ĐẦU

Việc ứng dụng Công nghệ thông tin hiện nay đã trở nên phổ biến trong mọi lĩnh vực của đời sống xã hội. Ngành Công nghệ thông tin luôn thu hút được nhiều học sinh, sinh viên theo học. Cơ sở dữ liệu là học phần bắt buộc đối với sinh viên theo học bất cứ ngành nào thuộc lĩnh vực Công nghệ thông tin. Để có thể hiểu rõ và vận dụng được những kiến thức đã học vào thực tế sinh viên cần phải thực hành với cơ sở dữ liệu. Chính vì thế, các chương trình đào tạo trong lĩnh vực Công nghệ thông tin thường thiết kế học phần Thực hành Cơ sở dữ liệu là học phần song hành với học phần Cơ sở dữ liệu. Như vậy, bên cạnh việc tìm hiểu lý thuyết về Cơ sở dữ liệu, sinh viên có cơ hội thực hành trên máy tính, nhờ đó sinh viên hiểu sâu hơn về những kiến thức lý thuyết đã học, thấy được mối liên hệ với thực tế.

Với kinh nghiệm giảng dạy và kiến thức tích lũy qua nhiều năm, nhận thấy sự cần thiết của việc thực hành cơ sở dữ liệu, nhóm tác giả đã quyết định ra đời cuốn sách hướng dẫn Thực hành Cơ sở dữ liệu. Trong cuốn sách này, nhóm tác giả cố gắng trình bày một cách cô đọng và dễ hiểu nhất những kiến thức liên quan đến thực hành cơ sở dữ liệu. Hy vọng cuốn sách này sẽ là tài liệu hữu ích cho sinh viên cũng như những ai quan tâm đến Cơ sở dữ liệu. Cuốn sách này gồm có ba chương, mỗi chương đều có phần cơ sở lý thuyết và hướng dẫn thực hành trong đó có trình bày các khái niệm cơ bản, hướng dẫn thực hành và có ví dụ kèm theo. Ngoài ra, có phần bài tập cho sinh viên thực hiện trên lớp và về nhà.

Chương 1: Cài đặt cơ sở dữ liệu

Chương này nhóm tác giả giới thiệu về cách thiết lập mô hình thực thể kết hợp, tạo các đối tượng trong cơ sở dữ liệu như: cơ sở dữ liệu, bảng, lược đồ Diagram, bảng ảo, nhập dữ liệu và thiết lập các ràng buộc về khóa.

Chương 2: Truy xuất dữ liệu.

Chương này giới thiệu về các câu lệnh truy vấn dữ liệu từ mức độ đơn giản đến phức tạp.

Chương 3: Cài đặt ràng buộc toàn vẹn.

Chương này hướng dẫn cách thiết lập và kiểm tra các ràng buộc toàn vẹn đối với cơ sở dữ liệu: ràng buộc Constraint và ràng buộc Trigger.

Cuốn sách chắc chắn không tránh khỏi những thiếu sót, chúng tôi rất mong nhận được ý kiến đóng góp từ các thầy giáo, cô giáo cũng như các bạn học sinh, sinh viên để cuốn sách được hoàn thiện hơn.

Nhóm tác giả

MỤC LỤC

CHƯƠNG 1. CÀI ĐẶT CƠ SỞ DỮ LIỆU	1
PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH	1
1.1. Giới thiệu các công cụ	1
1.2. Thiết kế mô hình thực thể kết hợp	2
1.3. Chuyển mô hình thực thể kết hợp sang mô hình quan hệ.....	14
1.3.1. Chuyển đổi từ mô hình CDM sang mô hình PDM	14
1.3.2. Chuyển đổi từ mô hình PDM sang SQL Server.....	18
1.4. Tạo cơ sở dữ liệu.....	24
1.5. Tạo bảng.....	25
1.6. Ràng buộc khoá chính, khoá ngoại	28
1.7. Tạo lược đồ Diagram	29
1.8. Nhập dữ liệu vào bảng	32
1.9. Tạo bảng ảo.....	34
PHẦN 2: BÀI TẬP THỰC HÀNH	36
PHẦN 3: BÀI TẬP LÀM THÊM	45
CHƯƠNG 2. TRUY XUẤT DỮ LIỆU.....	50
PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH	50
2.1. Cập nhật dữ liệu	51
2.1.1. Thêm dữ liệu	51
2.1.2. Xóa dữ liệu	51
2.1.3. Sửa dữ liệu	52
2.2. Truy vấn dữ liệu.....	53
2.2.1. Cú pháp câu truy vấn.....	53
2.2.2. Sử dụng tên bí danh, *	54
2.2.3. Xử lý chuỗi bằng toán tử Like	56
2.2.4. Các hàm thống kê	57
2.3. Truy vấn nâng cao.....	60
2.3.1. Truy vấn lòng	60
2.3.2. Truy vấn với các phép toán tập hợp (UNION, INTERSECT, EXCEPT).61	61

2.3.3. Truy vấn với phép kết ngoài (outer join)	64
2.3.3.1. Inner Join	65
2.3.3.2. Left outer Join.....	66
2.3.3.3. Right outer Join.....	68
2.3.3.4. Full outer Join	68
2.3.3.5. Một số trường hợp khác của câu lệnh truy vấn	69
PHẦN 2: BÀI TẬP THỰC HÀNH	70
PHẦN 3: BÀI TẬP LÀM THÊM	76
CHƯƠNG 3. CÀI ĐẶT RÀNG BUỘC TOÀN VẸN	78
PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH	78
3.1. Khái niệm.....	78
3.2. Cài đặt ràng buộc toàn vẹn.....	79
3.2.1. Cài đặt ràng buộc toàn vẹn dùng Constraint	79
3.2.2. Cài đặt ràng buộc toàn vẹn dùng Trigger.....	84
PHẦN 2: BÀI TẬP THỰC HÀNH	90
PHẦN 3: BÀI TẬP LÀM THÊM	91
TÀI LIỆU THAM KHẢO	93

DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

CSDL: Cơ sở dữ liệu

DDL: Data Definition Language – Ngôn ngữ định nghĩa dữ liệu

DML: Data Manipulation Language – Ngôn ngữ thao tác dữ liệu

FK: Foreign key – Khoá ngoại

PK: Primary Key – Khoá chính

RBTY: Ràng buộc toàn vẹn

SQL: Structured Query Language – Ngôn ngữ truy vấn cấu trúc

T-SQL: Transact Structured Query Language

DANH MỤC HÌNH ẢNH

Hình 1.1. Màn hình đăng nhập SQL Server 2012	24
Hình 1.2. Màn hình tạo cơ sở dữ liệu	25
Hình 1.3. Màn hình tạo khoá chính	25
Hình 1.4. Màn hình lựa chọn các thao tác trên bảng	27
Hình 1.5. Màn hình tạo liên kết trong lược đồ Diagram	31
Hình 1.6. Màn hình chọn chức năng trên bảng trong lược đồ Diagram.....	31
Hình 1.7. Màn hình chọn chức năng nhập dữ liệu trực tiếp trên bảng	32
Hình 1.8. Màn hình nhập dữ liệu trực tiếp trên bảng	32
Hình 1.9. Màn hình chọn bảng tạo bảng ảo.....	35
Hình 1.10. Màn hình tạo bảng ảo	35
Hình 2.1. Màn hình giới thiệu trình soạn thảo lệnh SQL	51
Hình 2. 2. Mô tả phép kết nối Join	65
Hình 3.1. Giao diện quản lý Trigger.....	88
Hình 3.2. Giao diện xem nội dung Trigger	89
Hình 3. 3. Giao diện xoá Trigger.....	90

Trường ĐH CNTP TP.HCM
Khoa: CNTT
Bộ môn: HTTT
MSMH: 01201009

CHƯƠNG 1
CÀI ĐẶT CƠ SỞ DỮ LIỆU



MỤC TIÊU

- Thiết kế mô hình thực thể kết hợp, chuyển mô hình thực thể kết hợp sang mô hình quan hệ.
- Tạo các đối tượng trong cơ sở dữ liệu SQL Server 2012 như: Table, View, Diagram.
- Thiết lập ràng buộc khoá chính và khoá ngoại trên bảng.
- Nhập liệu vào các bảng trong cơ sở dữ liệu.

DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SINH VIÊN

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Máy tính cài đặt SQL Server 2012	1	1	

NỘI DUNG THỰC HÀNH

PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH

1.1. Giới thiệu các công cụ

Để thiết kế các mô hình cơ sở dữ liệu có nhiều công cụ hỗ trợ. Trong nội dung này, tài liệu giới thiệu một số công cụ thiết kế mô hình thực thể kết hợp thông dụng, cụ thể:

- ❖ **Power Designer:** là một phần mềm hỗ trợ thiết kế nhiều mô hình như: *Conceptual Data model* (CDM), *Physical Data Model* (PDM), *Business* (BPMN, BPEL, ebXML), *Application* (UML),..., phần mềm hỗ trợ phát sinh *Physical Data Model* (PDM).

Ưu điểm của DBDesigner:

- Thiết kế mô hình CDM.
- Chuyển đổi từ mô hình CDM sang PDM.
- Chuyển đổi từ mô hình PDM sang mô hình dữ liệu quan hệ.
- Phát sinh code tự động tạo cơ sở dữ liệu (bao gồm các ràng buộc) trên một DBMS đích.

- ❖ **Diagrams.net (Draw.io):** là công cụ phổ biến để thiết kế UML, ERD, Organization Chart, Flow Char,... chạy online không cần cài đặt.

Ưu điểm của Diagrams.net:

- Thiết kế diagram miễn phí

- Cho phép cộng tác trực tuyến qua Google Drive theo quyền hạn: Editor, Viewer, Commenter.
 - Có nhiều mẫu hình có sẵn.
 - Đường nét, màu sắc hình trong app.diagrams.net đa dạng để thiết kế ERD.
 - Hỗ trợ nhiều tab kiểu như worksheet trong Excel
 - Hỗ trợ paste vào lệnh tạo bảng CREATE TABLE để vẽ ra bảng.
 - Mức độ chi tiết, đầy đủ đường liên kết giữa hai bảng.
- ❖ **DBDesigner:** là một ứng dụng web thiết kế cho phép thiết kế cơ sở dữ liệu trực tuyến và là công cụ mô hình hóa cho MySQL, SQL Server, PostgreSQL, Oracle và SQLite.
- Ưu điểm của DBDesigner:*
- Tính năng chuyên nghiệp
 - Giao diện người dùng rõ ràng và đơn giản

Ngoài những công cụ trên, hiện nay còn nhiều công cụ khác được sử dụng hỗ trợ thiết kế mô các mô hình cơ sở dữ liệu như: Lucidchart, DbSchema, SmartDraw,...

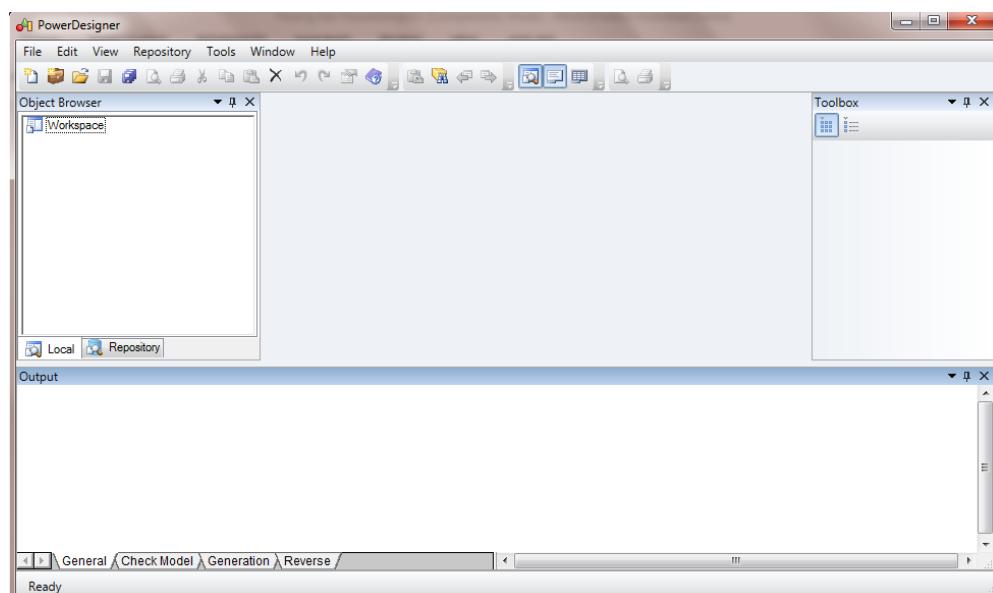
1.2. Thiết kế mô hình thực thể kết hợp

Mô hình dữ liệu quan niệm (Conceptual Data Model – CDM) là một dạng mô hình dùng để biểu diễn các đối tượng dữ liệu trong một hệ thống. Có nhiều dạng mô hình quan niệm dữ liệu như: mô hình dữ liệu mạng, mô hình dữ liệu phân cấp, mô hình thực thể kết hợp,... Trong đó, mô hình thực thể kết hợp là một trong những mô hình biểu diễn dữ liệu thông dụng nhất.

Trong phần này, tài liệu sử dụng phần mềm Power Designer 16 để minh họa thiết kế mô hình thực thể kết hợp.

❖ Khởi động Power Designer 16

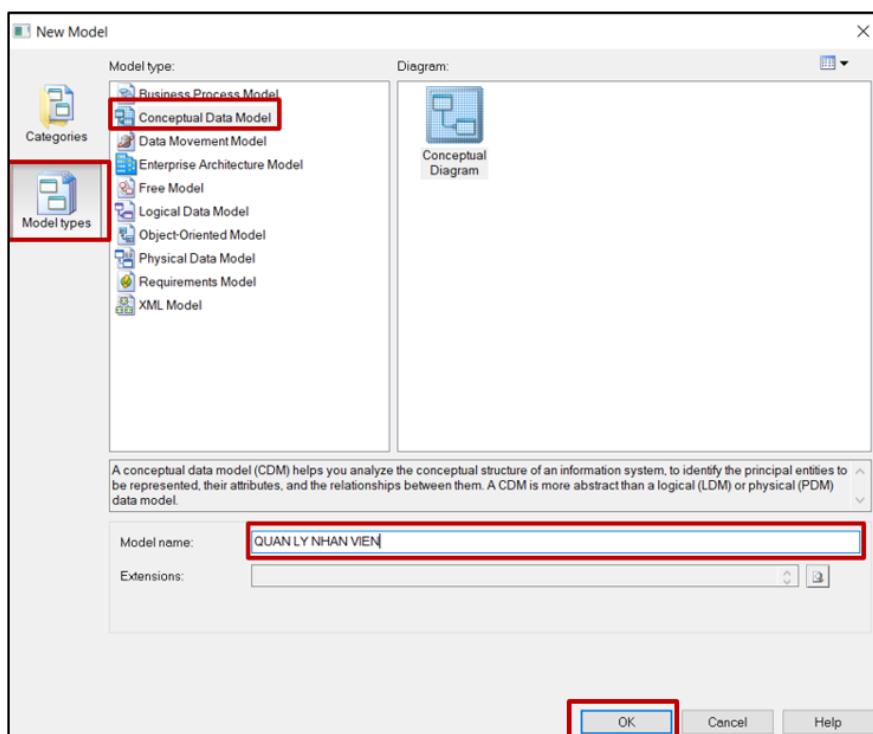
Start/All Programs/ Sybase/ PowerDesigner 16/ PowerDesigner



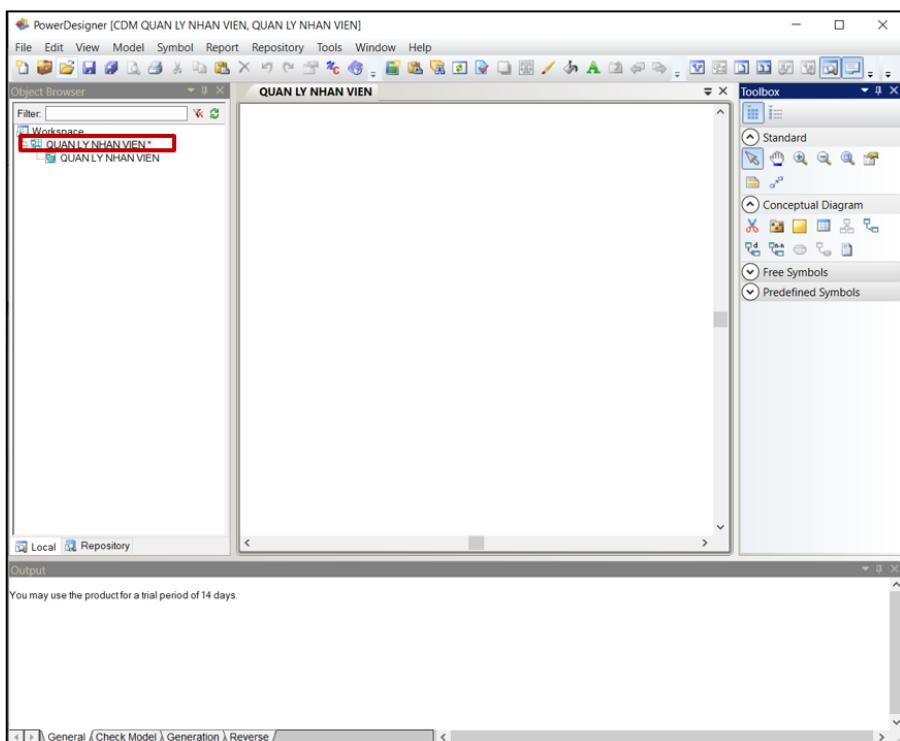
❖ **Tạo mới một mô hình quan niệm dữ liệu**

File/ New Model. Hộp thoại New Model xuất hiện.

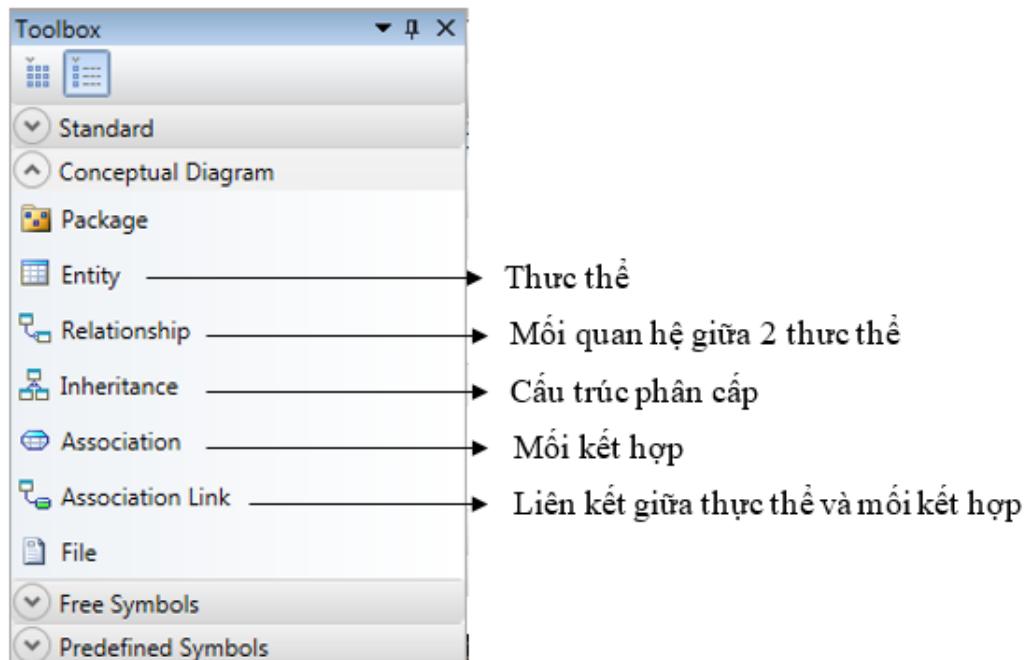
- ✓ Chọn Model types là Conceptual Data Model.
- ✓ Gõ tên mô hình vào ô Model name.
- ✓ Click nút OK.



Tên mô hình xuất hiện trong cửa sổ Workspace

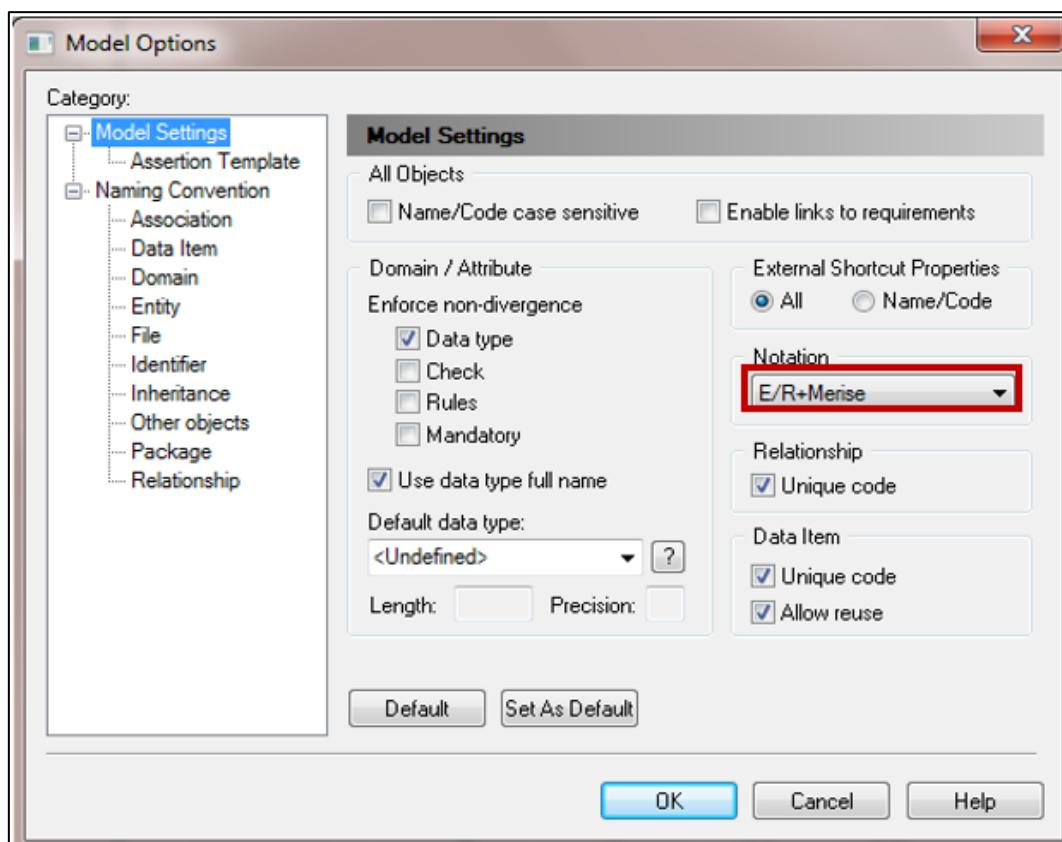


❖ Thanh công cụ



Cách bật Association và Association link:

Vào menu: tools-> Model options-> Notation, chọn "E/R+Merise"



❖ Thiết lập mô hình CDM

Ví dụ thiết lập mô hình CMD cho hệ thống quản lý nhân viên của một công ty với các thông tin cần quản lý các thông tin sau:

Công ty có nhiều phòng ban mỗi phòng ban có nhiều nhân viên. Thông tin phòng ban gồm: mã phòng, tên phòng, địa điểm. Thông tin nhân viên gồm: Mã nhân viên, họ tên, ngày sinh, phái, địa chỉ, lương của nhân viên. Thông tin đề án gồm: Mã đề án, tên đề án, địa điểm đề án, ngày bắt đầu. Mỗi nhân viên được phân công tham gia một hoặc nhiều đề án, mỗi đề án nhân viên được phân công có số ngày nhân viên tham gia. Mỗi nhân viên có một hoặc nhiều thân nhân phụ thuộc, mỗi thân nhân có thông tin: tên thân nhân, phái, ngày sinh, mối quan hệ với nhân viên. Ngoài ra mỗi nhân viên được cấp một thẻ nhân viên, thông tin thẻ gồm: mã thẻ, ngày cấp.

PHONGBAN

Thuộc tính	Kiểu dữ liệu	Kích thước	Ghi chú
MAPHONG	Character	10	Thuộc tính khóa
TENPHONG	Character	30	
DIADIEMPB	Character	40	

NHANVIEN

Thuộc tính	Kiểu dữ liệu	Kích thước	Ghi chú
MANV	Character	6	Thuộc tính khóa
HOTEN	Character	40	
NGAYSINH	Date		
PHAI	Character	5	
DIACHI	Character	40	
LUONG	Money		

DEAN

Thuộc tính	Kiểu dữ liệu	Kích thước	Ghi chú
MADA	Character	5	Thuộc tính khóa
TENDA	Character	50	
DIADIEMDA	Character	30	
NGAYBD	date		

THANHAN

Thuộc tính	Kiểu dữ liệu	Kích thước	Ghi chú
TENTN	Character	40	Thuộc tính khóa
PHAI	Character	5	
NGAYSINH	Date		
QUANHE	Character	15	

THENHANVIEN

Thuộc tính	Kiểu dữ liệu	Kích thước	Ghi chú
MATHE	Character	10	Thuộc tính khóa
NGAYCAP	Date		

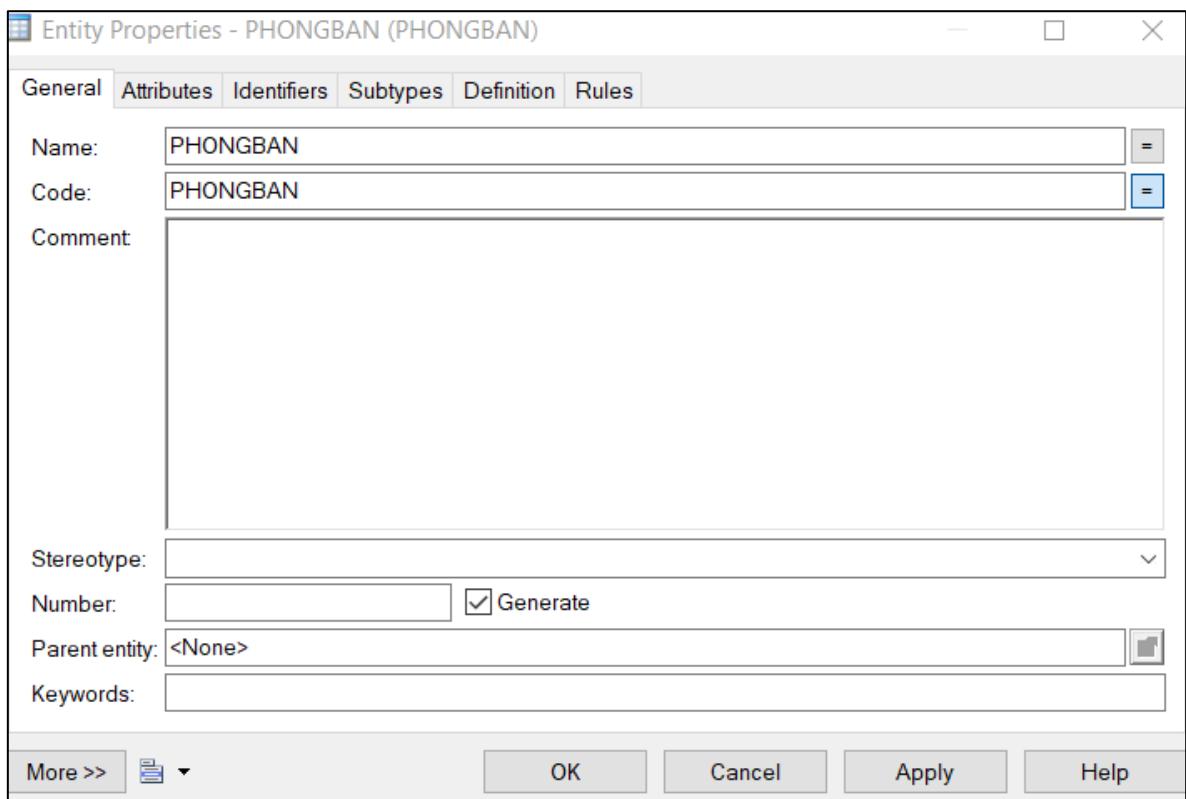
❖ Tạo thực thể

- ✓ Chọn biểu tượng thực thể trên Toolbox, sau đó click vào nơi muốn đặt thực thể trong lược đồ.
- ✓ Khai báo thông tin của thực thể:

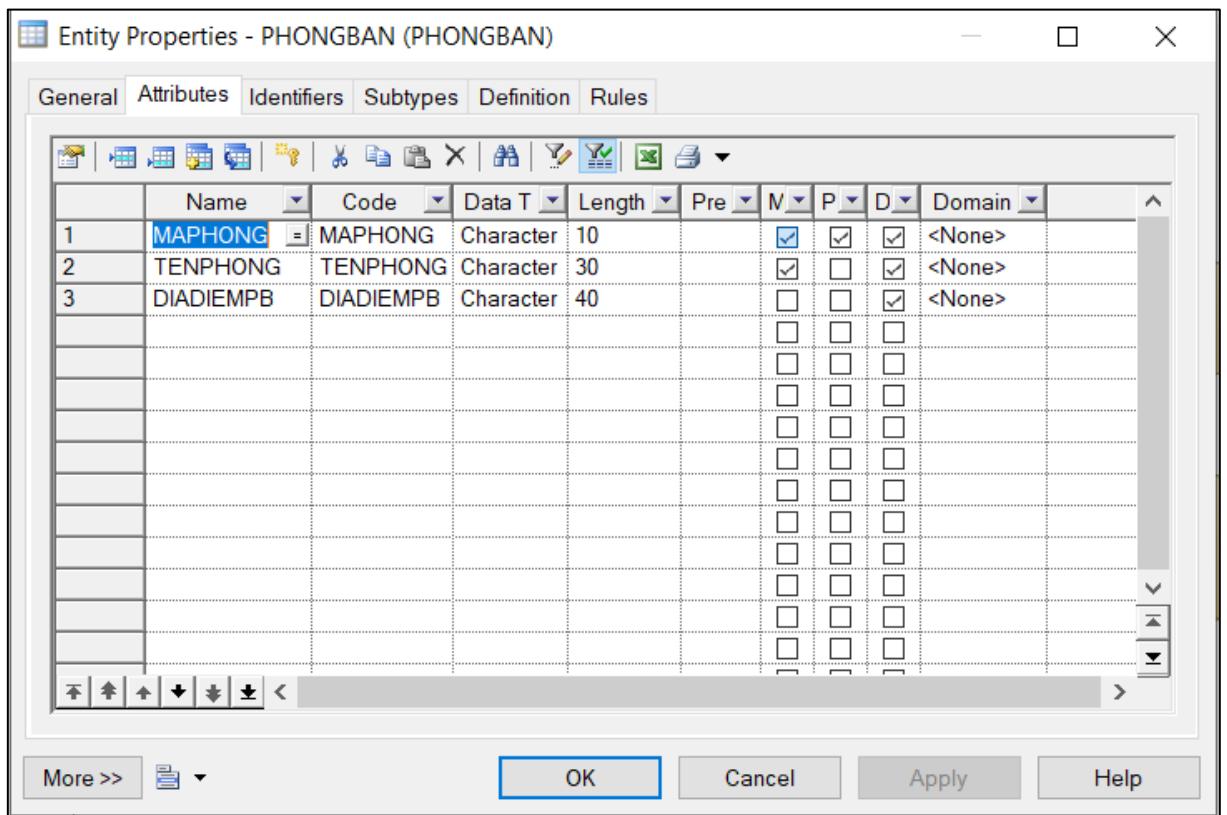
Double-click vào thực thể, một cửa sổ mới mở ra cho phép chúng ta chỉnh sửa thông tin của thực thể như: tên của thực thể, thuộc tính của thực thể, các rule,....

Thẻ General:

Name	Tên thực thể. Tên này hiển thị trên mô hình
Code	Tên tắt của thực thể. Tên này được dùng khi chuyển sang CSDL vật lý
Comment	Điễn giải về thực thể
Number	Số mẫu tin sẽ lưu trữ trong thực thể
Generate table	Được chọn nếu entity sẽ được chuyển thành table trong PDM



Thẻ Attributes : Khai báo thuộc tính của thực thể



- Name: nhập tên thuộc tính của thực thể.
- Code: phát sinh một mã tương ứng một thuộc tính.

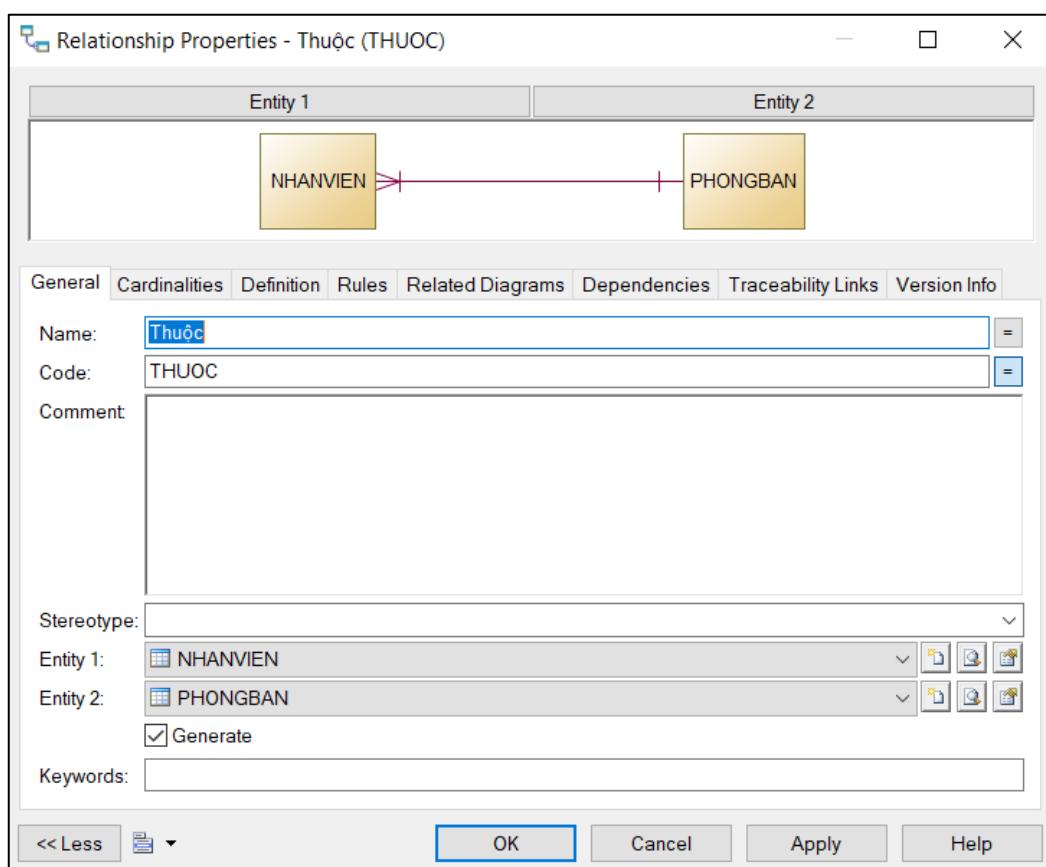
- Data Type: chọn kiểu dữ liệu cho thuộc tính, click vào dấu “...” bên cạnh, cửa sổ các kiểu dữ liệu mở ra cho phép chọn kiểu dữ liệu phù hợp cho các thuộc tính.
- M (Mandatory): thuộc tính bắt buộc/không bắt buộc.
- P(Primary Identifier): thuộc tính là khóa chính/không là khóa chính.
- D(Displayed): hiển thị thuộc tính/không hiển thị.

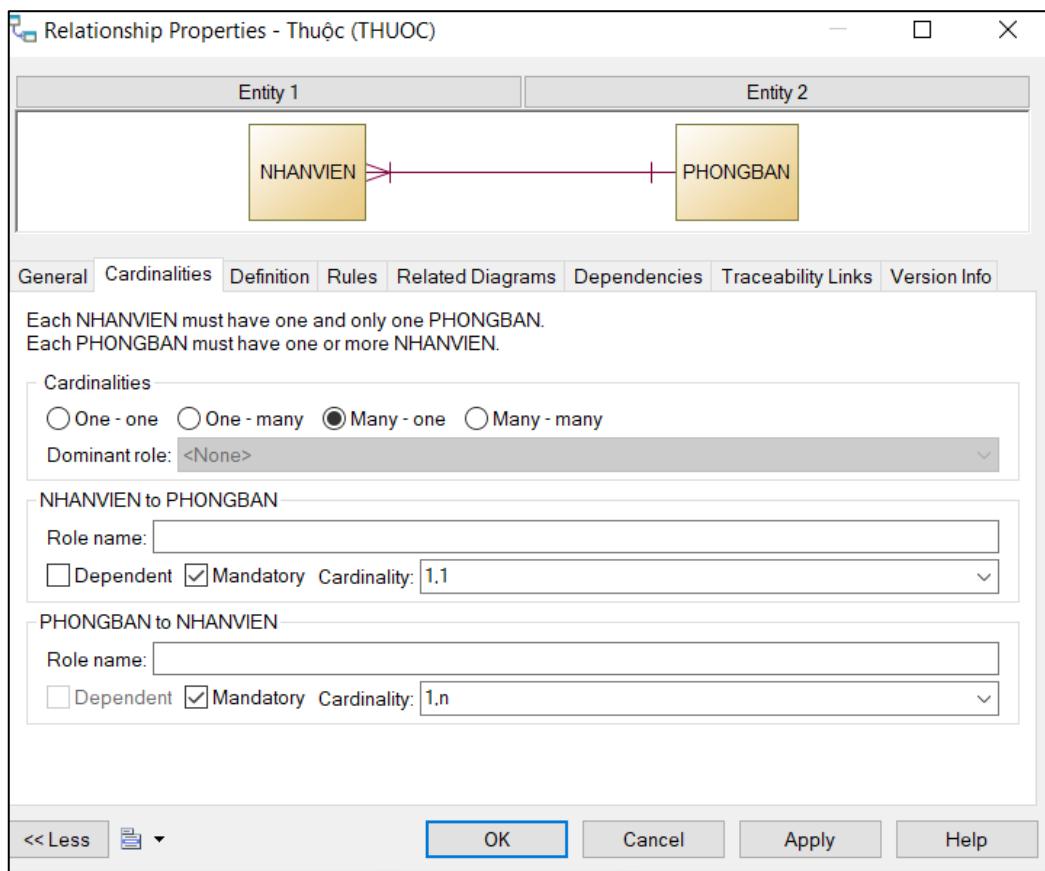
❖ Tạo mối kết hợp

- ✓ **Cách 1:** không sử dụng biểu tượng mối kết hợp ở giữa

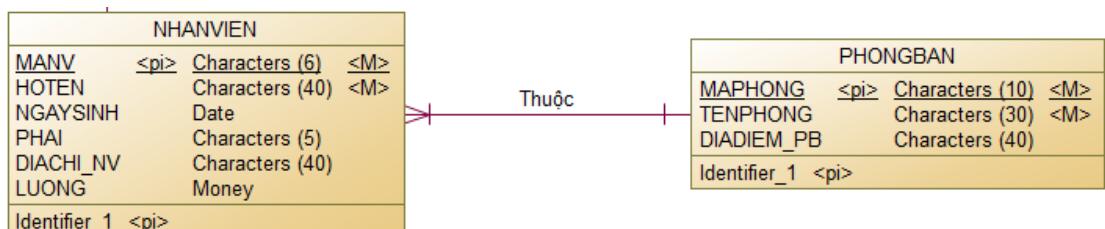
Giả sử xét mối quan hệ giữa hai thực thể PHONG và NHANVIEN

- Chọn biểu tượng mối quan hệ của hai thực thể  từ thanh công cụ.
- Click vào thực thể NHANVIEN và kéo qua thực thể PHONGBAN.
- Để thay đổi thông tin của mối kết hợp: như tên của mối kết hợp, bản số của mối kết hợp (1-1, 1-n, n-1,) double click vào mối quan hệ:





- Sau khi tạo xong, kết quả như sau:

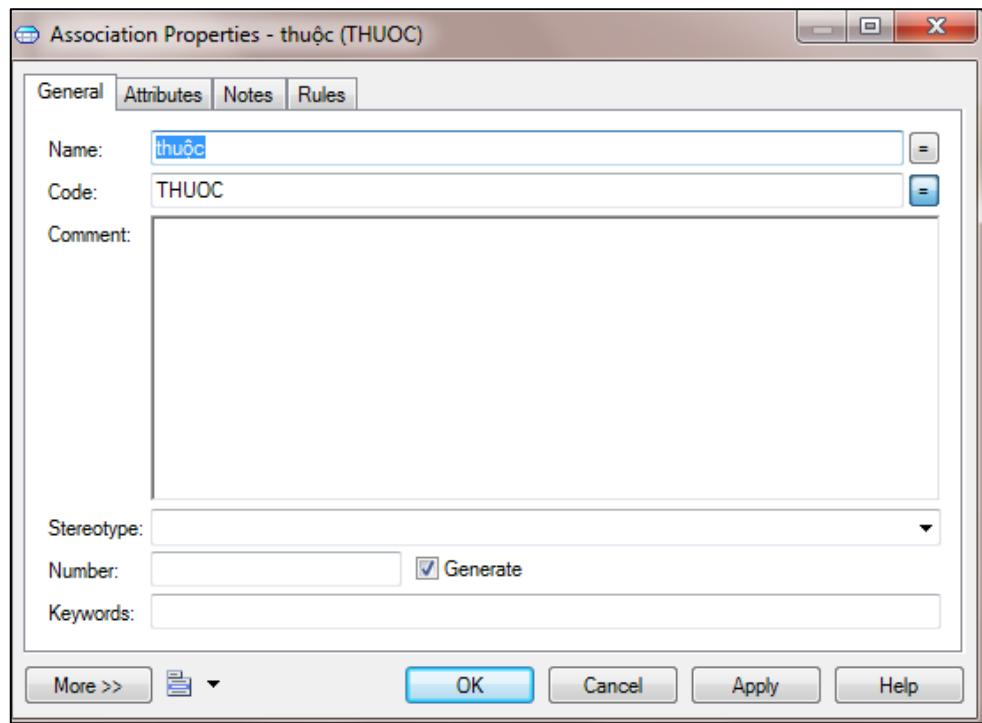


- ✓ **Cách 2:** sử dụng biểu tượng mối kết hợp ở giữa

Bước 1: tạo 1 mối kết hợp trong lược đồ

- Chọn biểu tượng Association từ thanh công cụ và đặt vào trong lược đồ.
- Double Click vào mối kết hợp vừa tạo: khai báo thông tin của mối kết hợp giống như khai báo thông tin của thực thể.

Lưu ý: *mối kết hợp không có thuộc tính khóa*

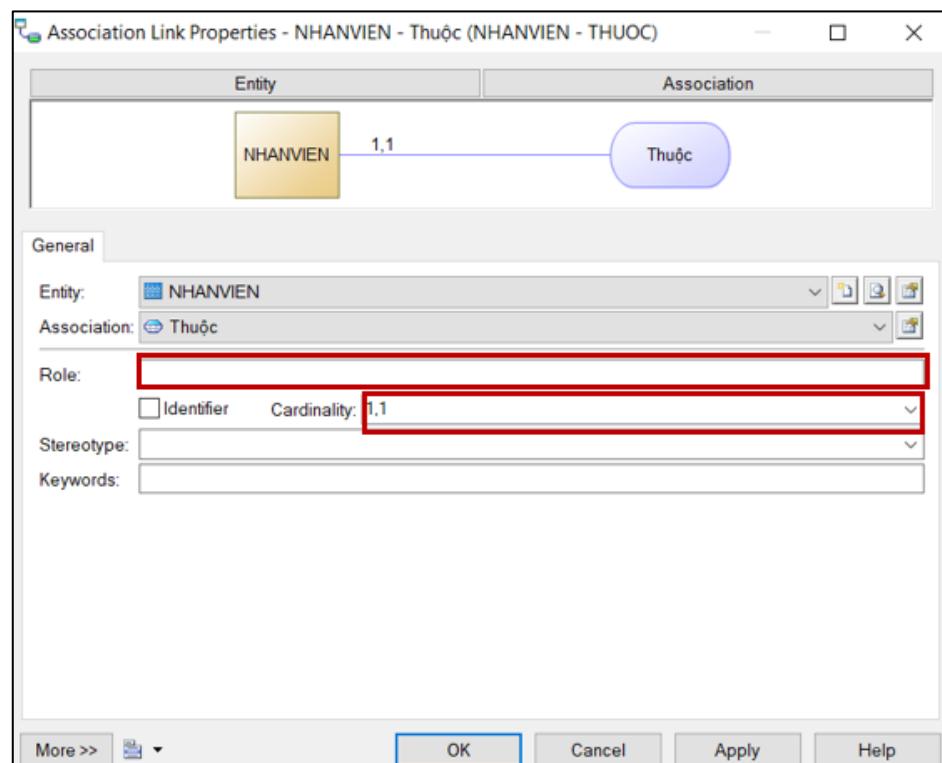


Bước 2: Tạo liên kết giữa thực thể với mối kết hợp

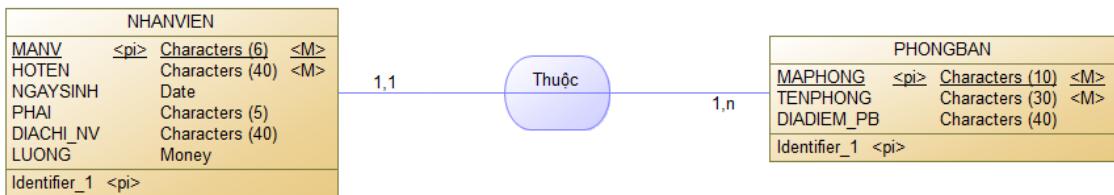
Click chọn biểu tượng Link, kéo thả từ thực thể đến mối kết hợp.

Chỉnh sửa các thông tin của liên kết:

- Role: diễn giải vai trò của mối liên kết
- Cardinality: Bản số mỗi nhánh của mối kết hợp.

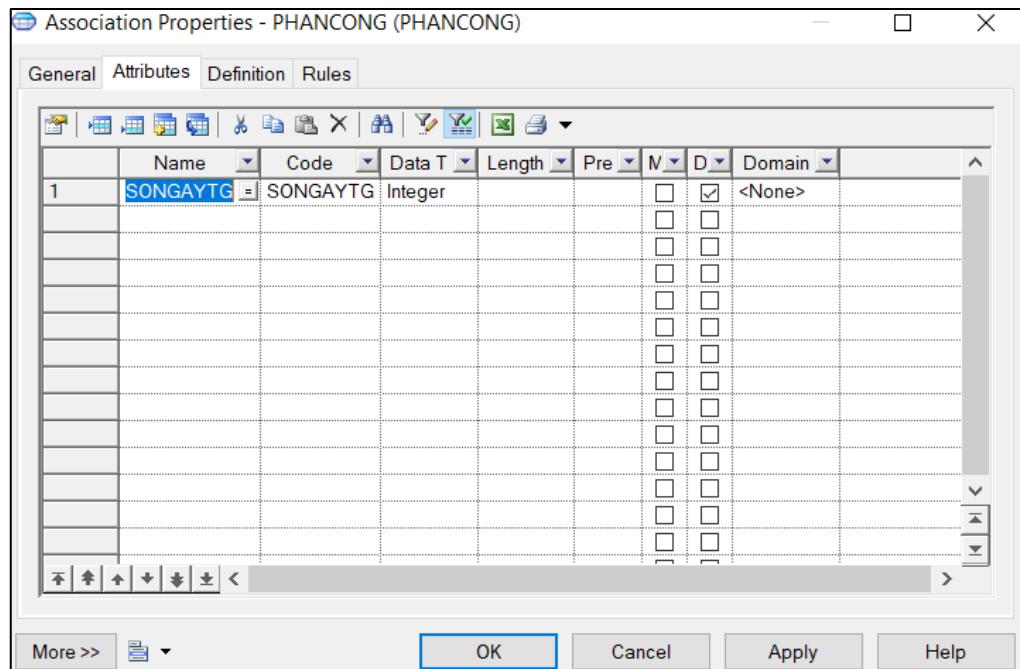


Kết quả sau khi tạo mối kết hợp:

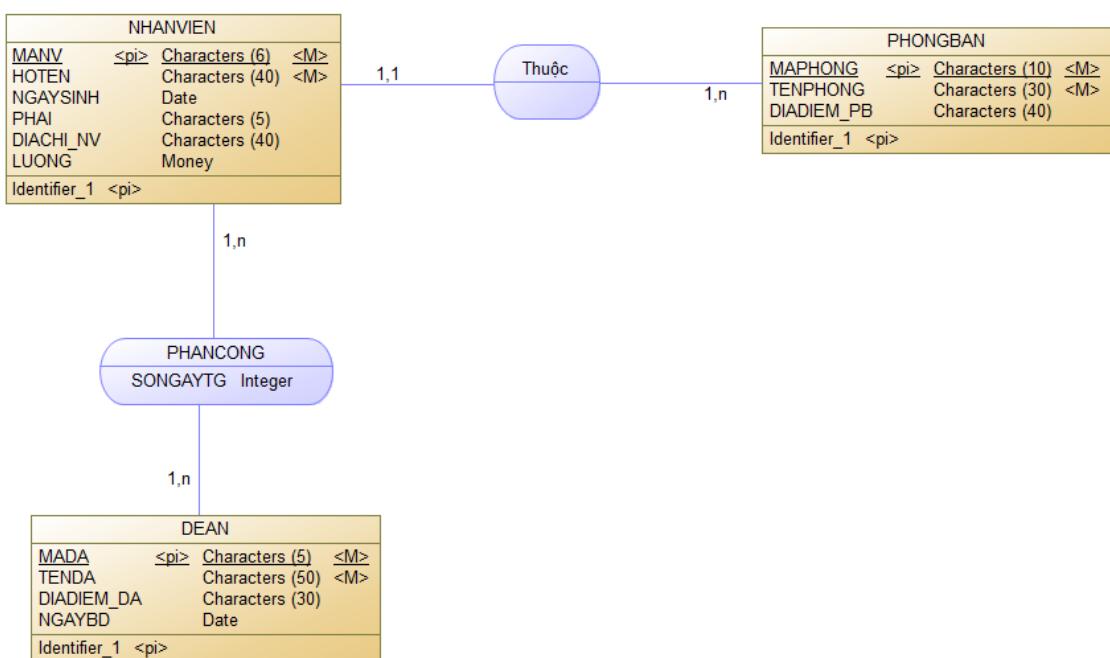


Tương tự, tạo mối kết hợp giữa thực thể NHANVIEN và DEAN:

Mối kết hợp giữa NHANVIEN và DEAN đặt tên là PHANCONG, thuộc tính Số ngày tham gia (SONGAYTG) thuộc về mối kết hợp này.



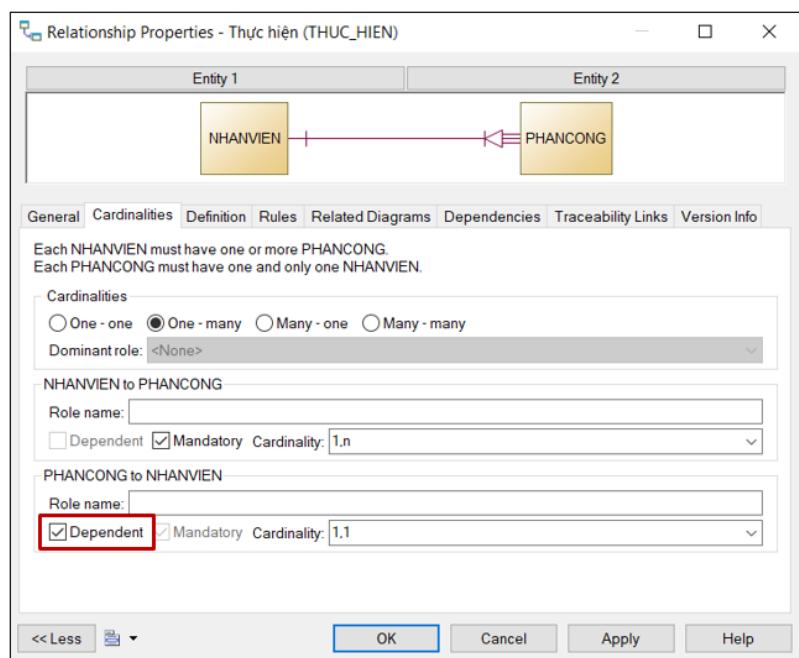
Kết quả sau khi tạo mối kết hợp:



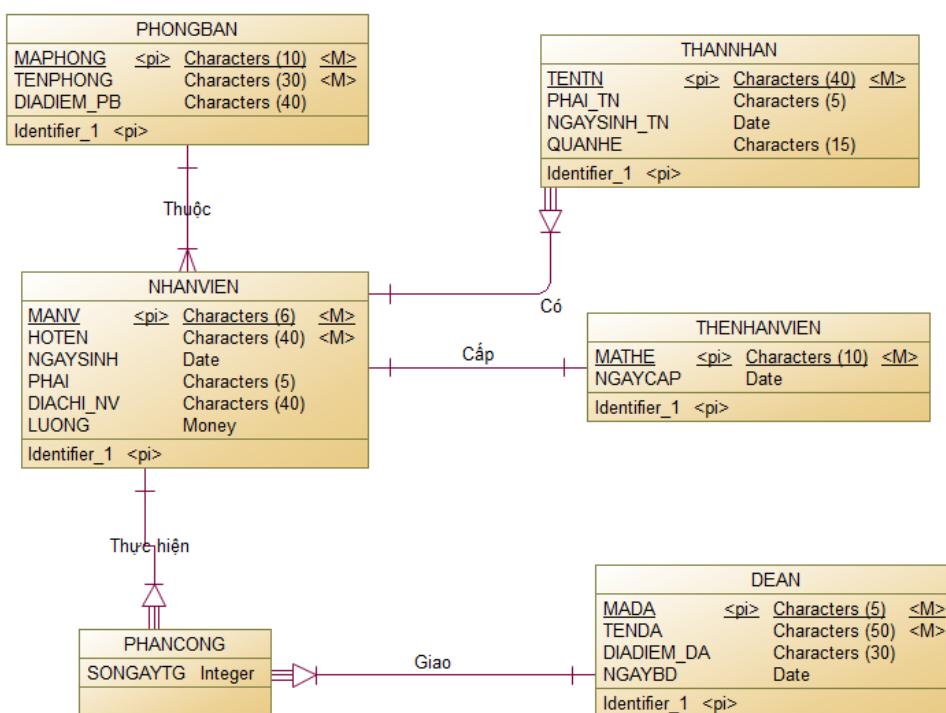
❖ Tạo thực thể yếu

Mỗi kết hợp PHANCONG trong ví dụ trên có thể chuyển thành một thực thể. Thuộc tính SONGAYTG trở thành thuộc tính của thực thể PHANCONG. Thực thể này là thực thể yếu, phụ thuộc vào 2 thực thể NHANVIEN và DEAN. Do đó, khóa của PHANCONG kế thừa từ khóa của hai thực thể NHANVIEN và DEAN.

Để một thực thể kế thừa khóa từ một thực thể khác, khi tạo mối quan hệ giữa hai thực thể, chọn *Dependent*:

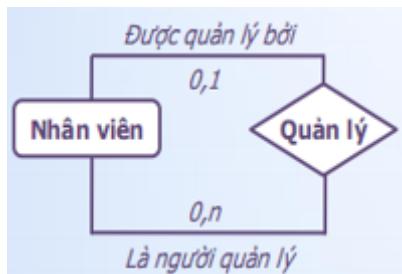


Kết quả sau khi tạo thực thể yếu:



❖ Biểu diễn mối kết hợp đệ quy

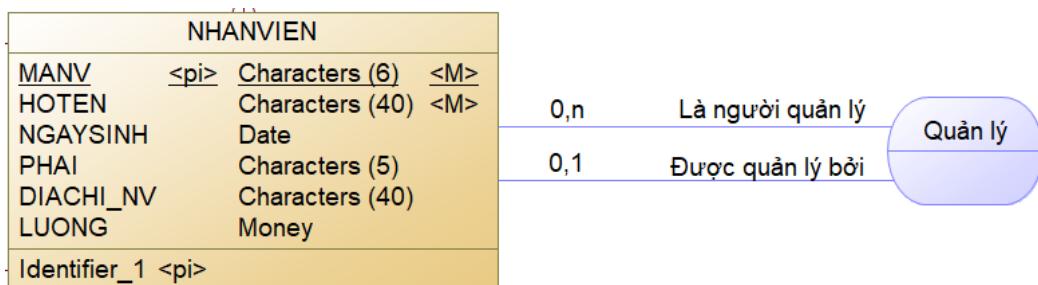
Giả sử ta có mối kết hợp đệ quy:



Cách tạo mối kết hợp đệ quy:

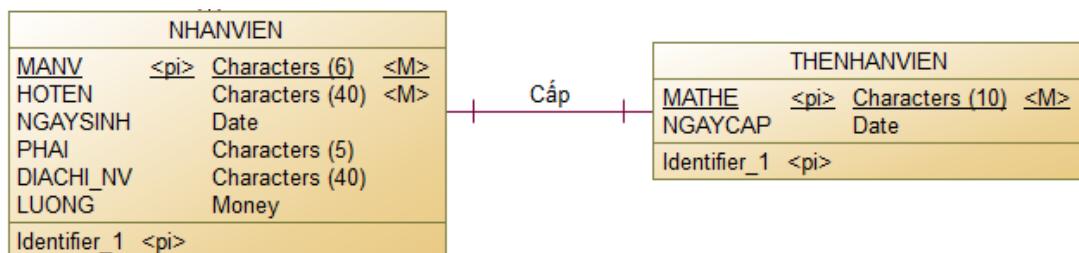
- **Bước 1:** vẽ thực thể Nhân viên
- **Bước 2:** vẽ mối kết hợp Quản lý
- **Bước 3:** tạo 2 liên kết giữa thực thể Nhân viên và mối kết hợp Quản lý

Kết quả sau khi tạo mối kết hợp đệ quy:

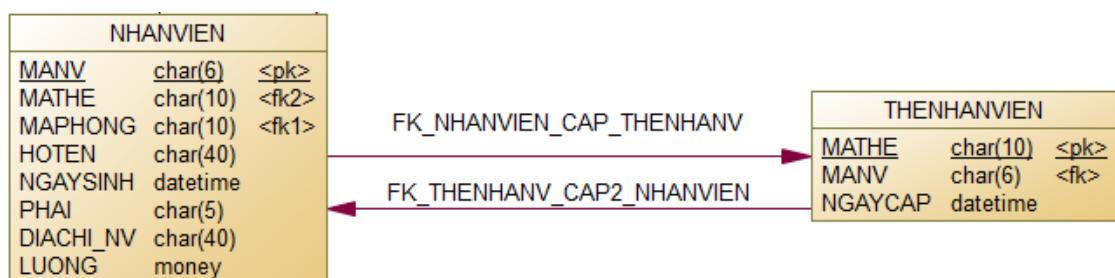


❖ Biểu diễn mối kết hợp 1-1

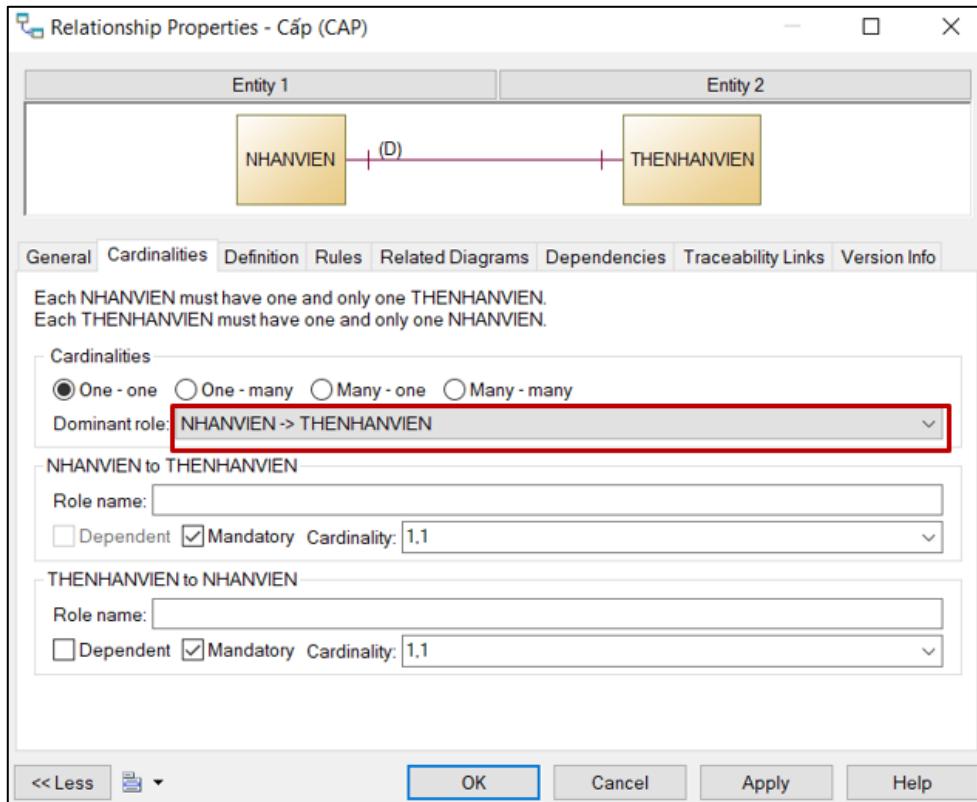
Bản số của mối kết hợp giữa 2 thực thể NHANVIEN và THENHANVIEN là 1-1 theo hình sau:



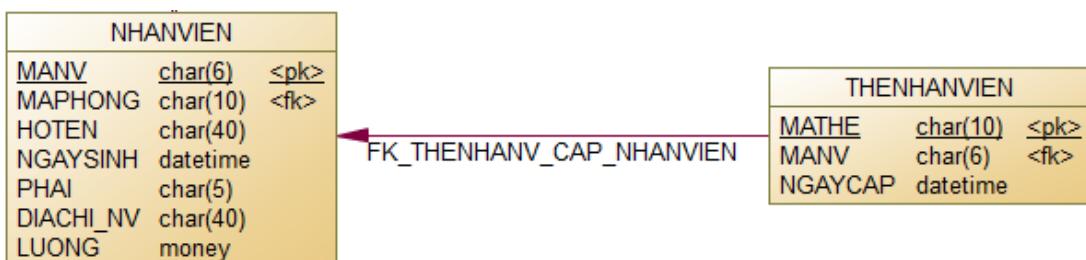
Khi chuyển sang PDM, mô hình có dạng:



Giả sử muốn đưa MANV vào table THENHANVIEN mà không muốn đưa MATHENV vào table NHANVIEN, *double click* vào đường link của mỗi kết hợp, chọn mục *Dominant role* như hình dưới:



Kết quả sau khi chuyển sang PDM như sau:



1.3. Chuyển mô hình thực thể kết hợp sang mô hình quan hệ

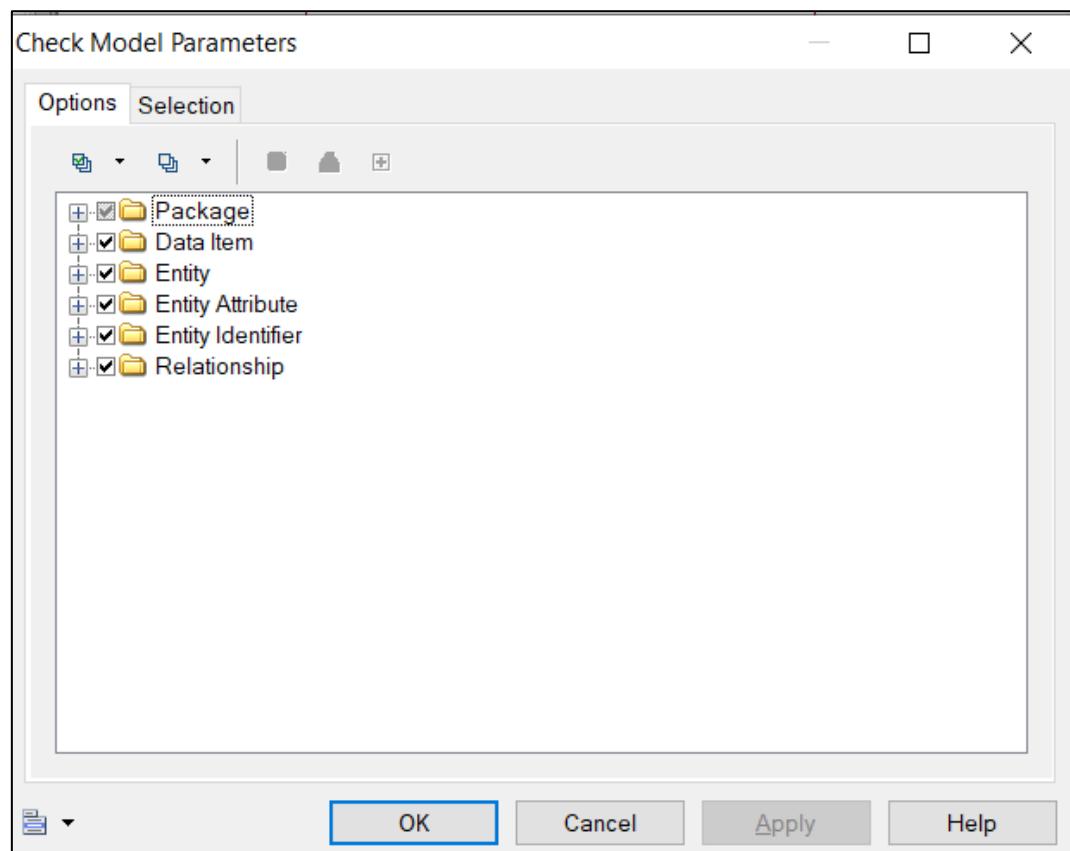
1.3.1. Chuyển đổi từ mô hình CDM sang mô hình PDM

Mô hình dữ liệu ở mức vật lý (*Physical Data Model - PDM*) là mô hình biểu diễn cấu trúc dữ liệu sẽ được cài đặt trong các hệ quản trị cơ sở dữ liệu như: SQL server, Oracle, Microsoft Word Access, DB2, MySQL,...

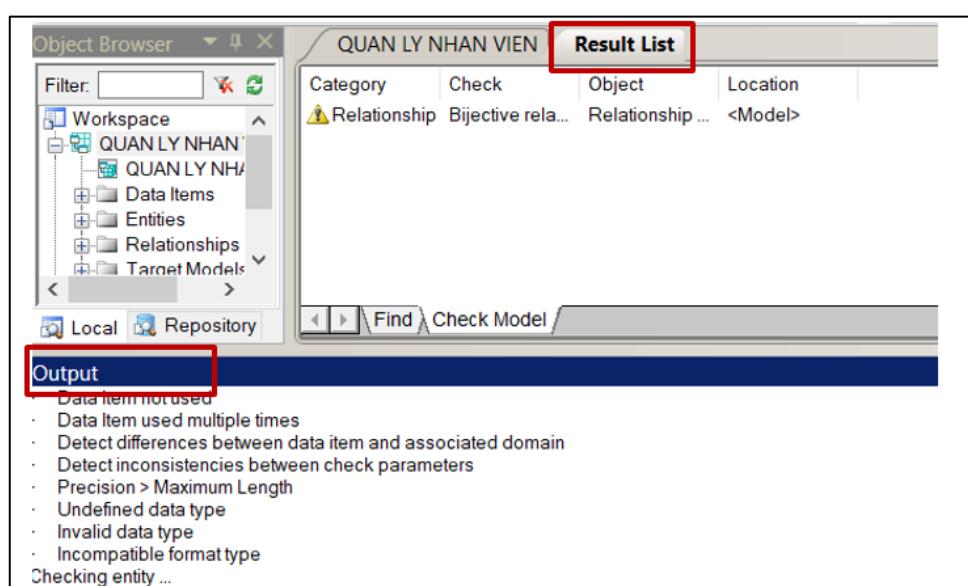
Các bước chuyển đổi từ mô hình CDM sang mô hình PDM:

Bước 1: Kiểm tra mô hình

- Chọn *Tool/Check Model* hoặc nhấn phím tắt F4 → hiển thị hộp *Check Conceptual Data Model*



- Lựa chọn những đối tượng cần kiểm tra lỗi. Sau đó, chọn OK để kiểm tra kết quả trong khung “Output”. Đối tượng và thuộc tính bị lỗi sẽ hiện trong khung Result List.



Trường hợp mô hình bị lỗi, kiểm tra chi tiết thông báo lỗi hoặc chuyển nhanh đến đối tượng bị lỗi bằng cách: click phải vào dòng thông báo trong hộp thoại “Result List” và chọn:

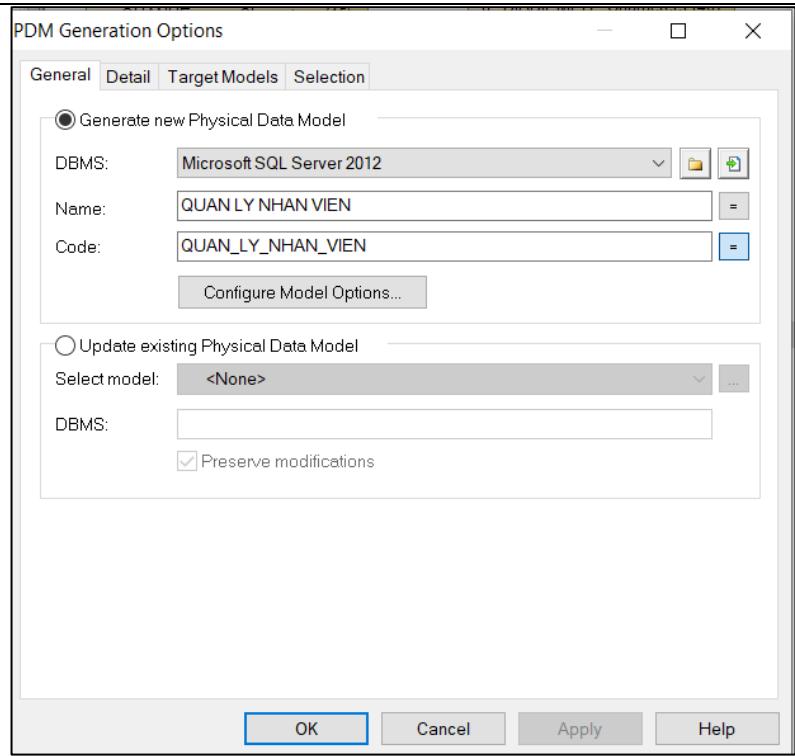
(1) *Detail* để xem chi tiết lỗi

(2) *Correct* mở cửa sổ thuộc tính của đối tượng bị lỗi để chỉnh sửa.

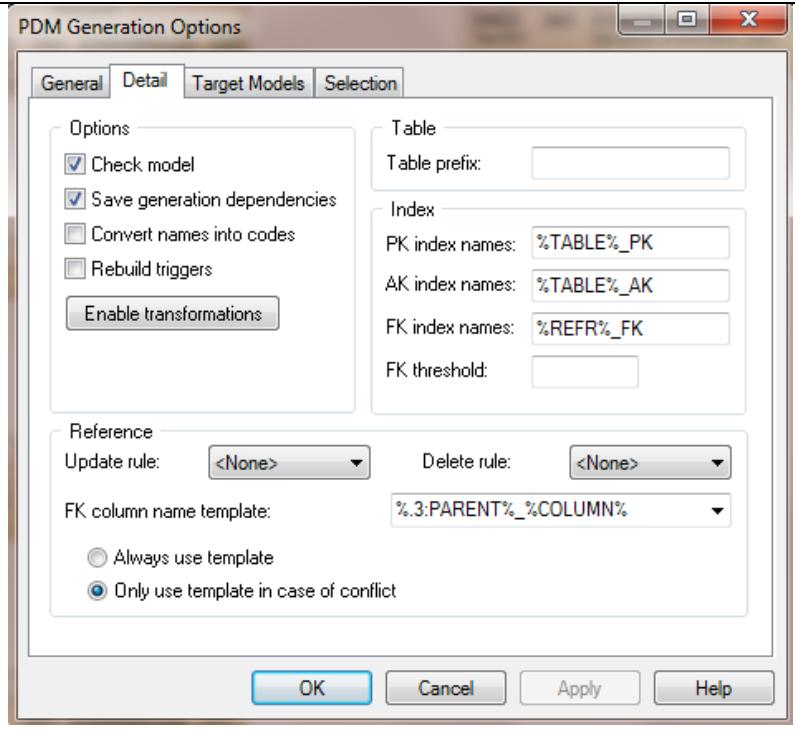
(3) *Recheck* để kiểm tra lại mô hình sau khi hiệu chỉnh

Bước 2: Chuyển mô hình quan niệm sang mô hình vật lý:

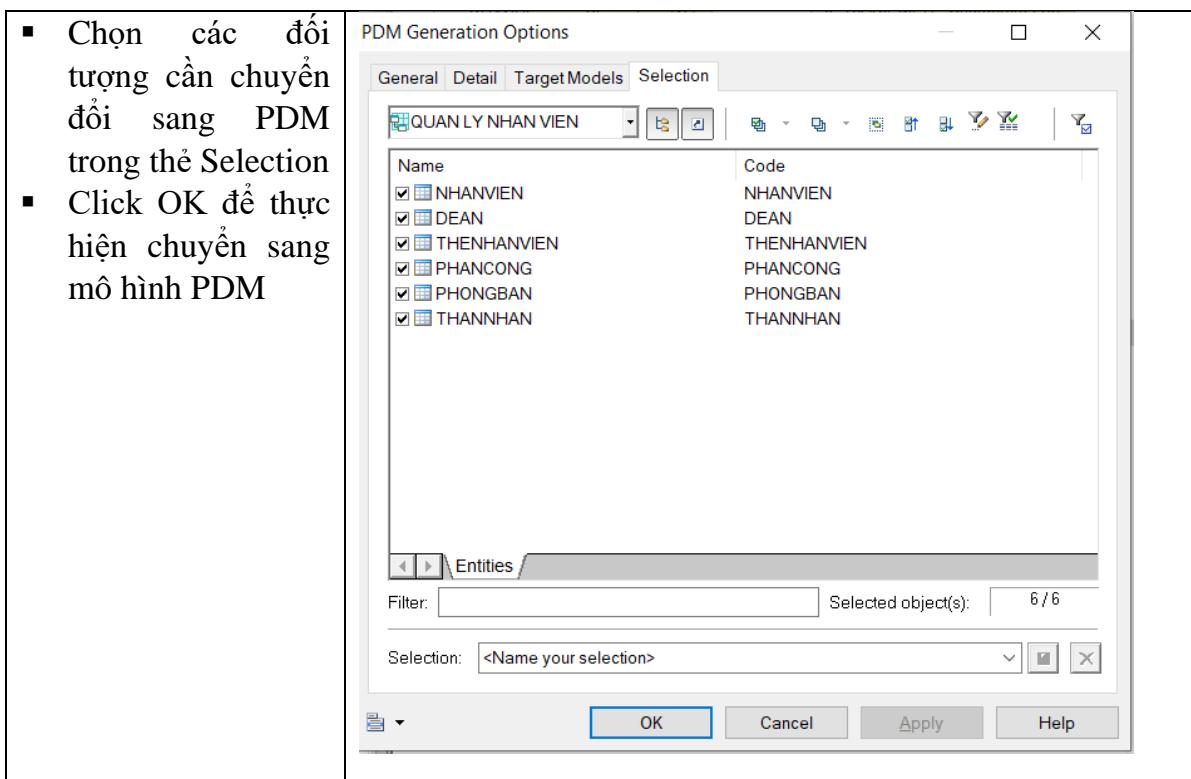
- Chọn **Tools → Generate Physical Data Model** hoặc sử dụng phím tắt **Ctrl+Shift+P**
- Chọn Hệ quản trị (DBMS) dùng để lưu trữ database trong thẻ **General**



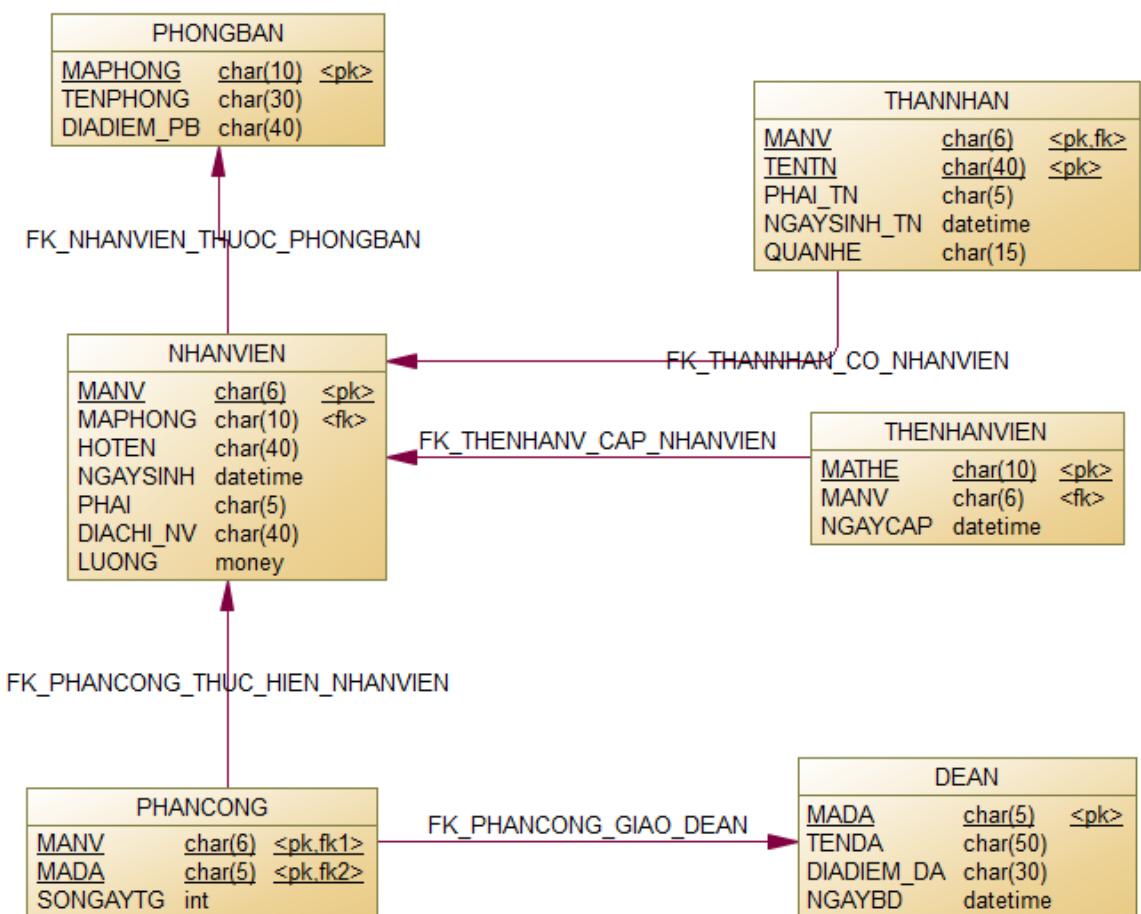
- Thiết lập các tùy chỉnh trong thẻ **Detail**



- Chọn các đối tượng cần chuyển đổi sang PDM trong thẻ Selection
- Click OK để thực hiện chuyển sang mô hình PDM



- Mô hình vật lý được tạo:



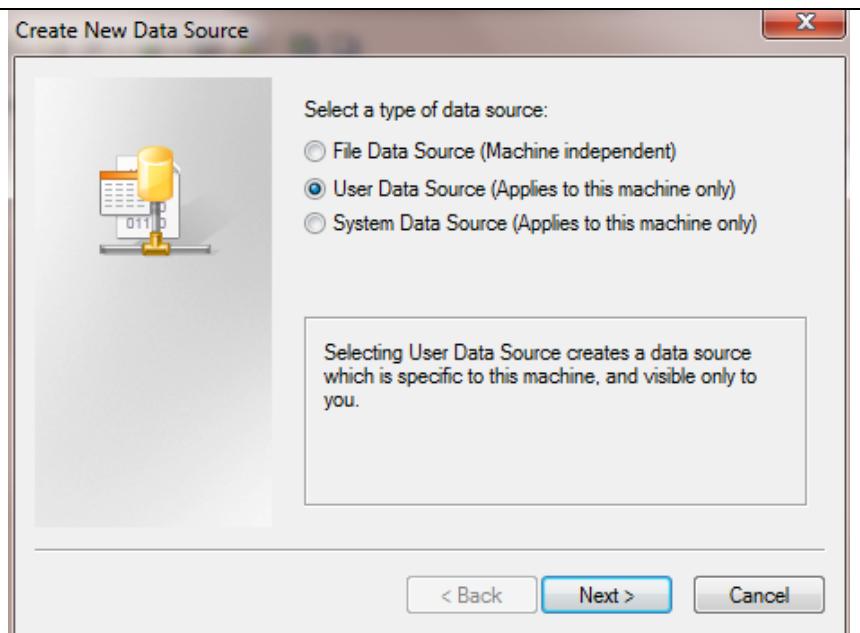
1.3.2. Chuyển đổi từ mô hình PDM sang SQL Server

- **Bước 1:** tạo CSDL mới trong SQL Server, giả sử Database mới trong SQL Server là QUANLY_NHANVIEN
- **Bước 2:** tạo kết nối từ Power Designer đến CSDL trên SQL Server.

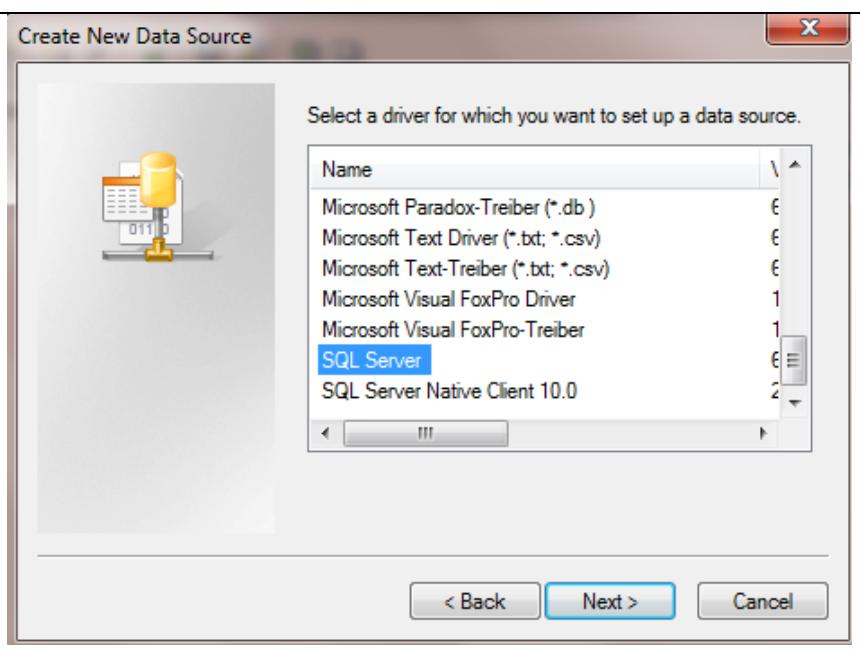
Tùy mô hình vật lý thực hiện:

(1) Chọn Database → Connect . Cửa sổ connect xuất hiện như hình bên.	
(2) Thực hiện kết nối qua một ODBC → chọn nút Configure tạo một kết nối ODBC, xuất hiện cửa sổ như hình bên. Chọn chức năng Add Data Source như trong hình.	

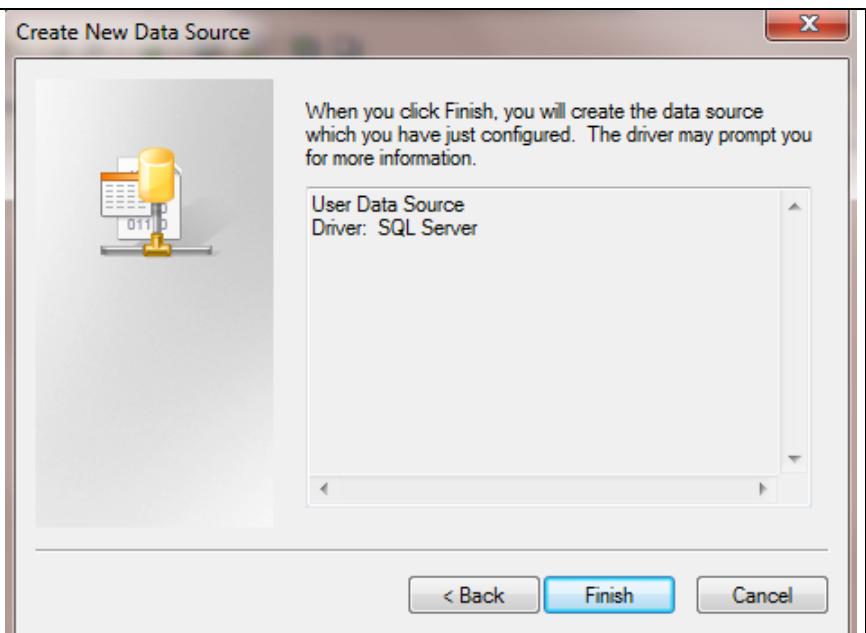
(3) Chọn loại *Data Source*, nhấn nút *Next*



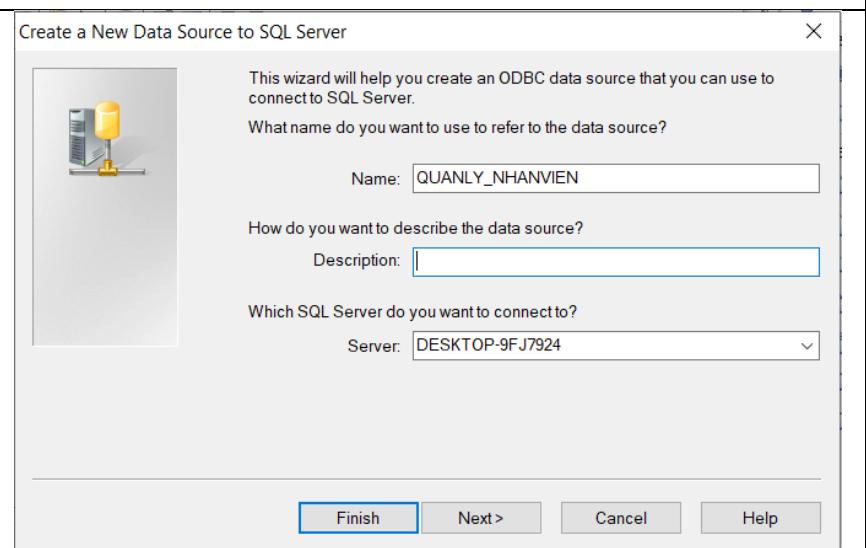
(4) Cửa sổ *Create New Data Source* xuất hiện, chọn *SQL Server* trong danh sách → *Next*



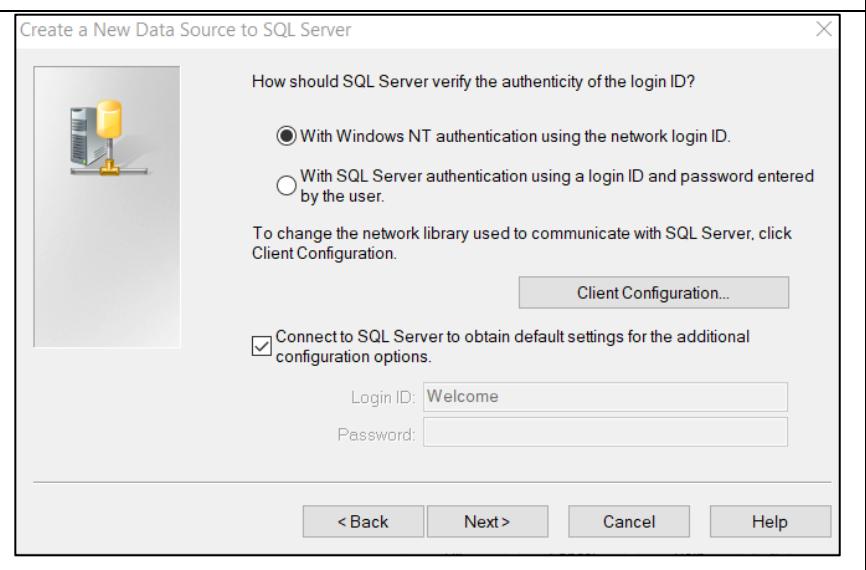
(5) Chọn **Finish**, xuất hiện cửa sổ **Create a new data source to SQL Server**



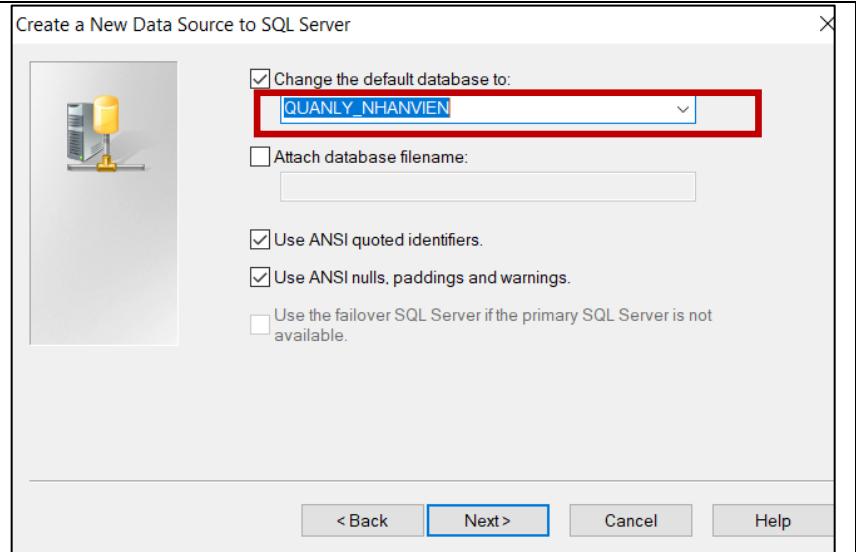
(6) Trong mục **Name** nhập tên CSDL, mục **Server** nhập vào tên Server đã kết nối bên → **Next**



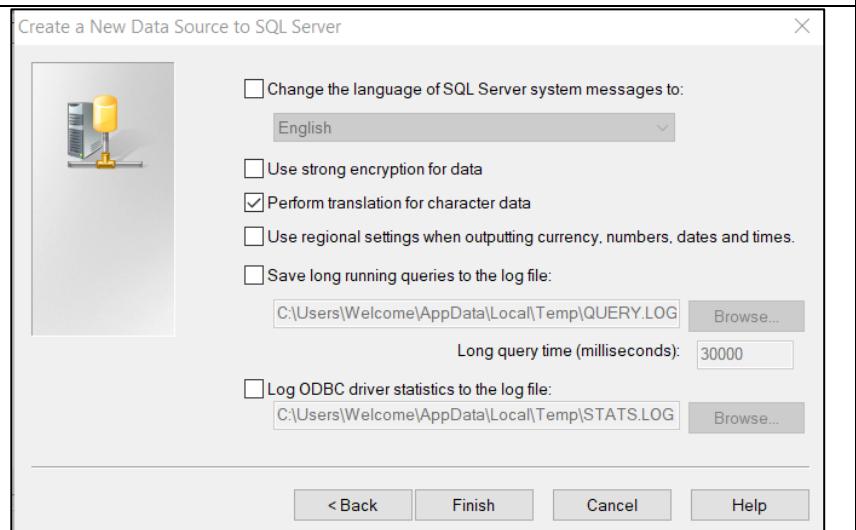
(7) Chọn kiểu kết nối đến SQL Server → chọn **Next**



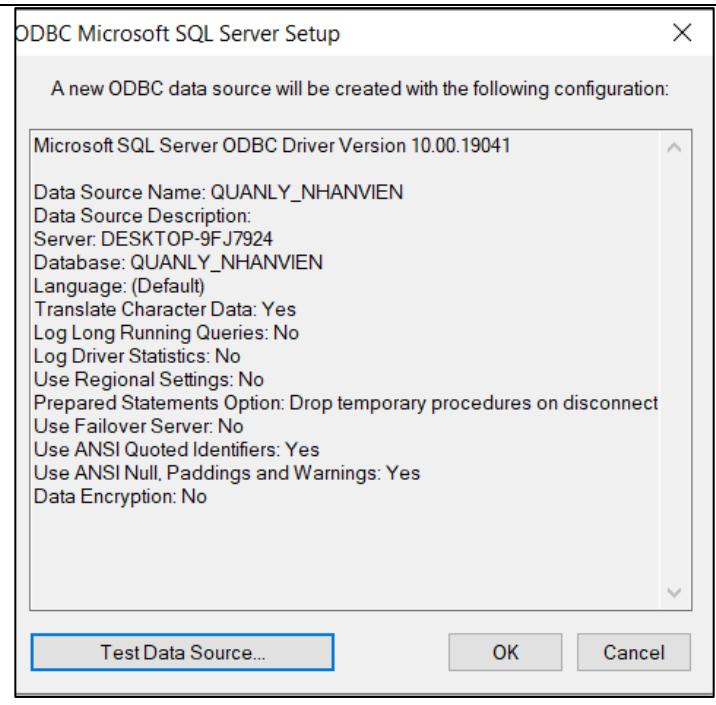
(8) Chọn CSDL cần kết nối → Chọn *Next*



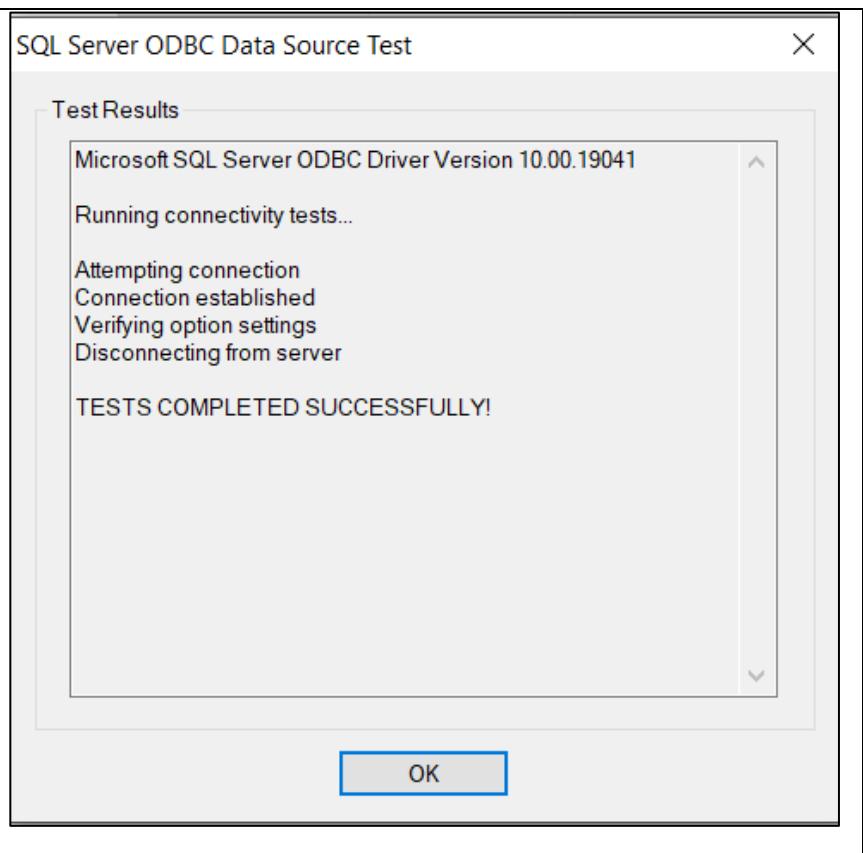
(9) Thiết lập các tùy chọn → chọn *Finish*.



(10) Cửa sổ *ODBC Microsoft SQL Server Setup* hiển thị như hình bên → chọn *Test Data Source*



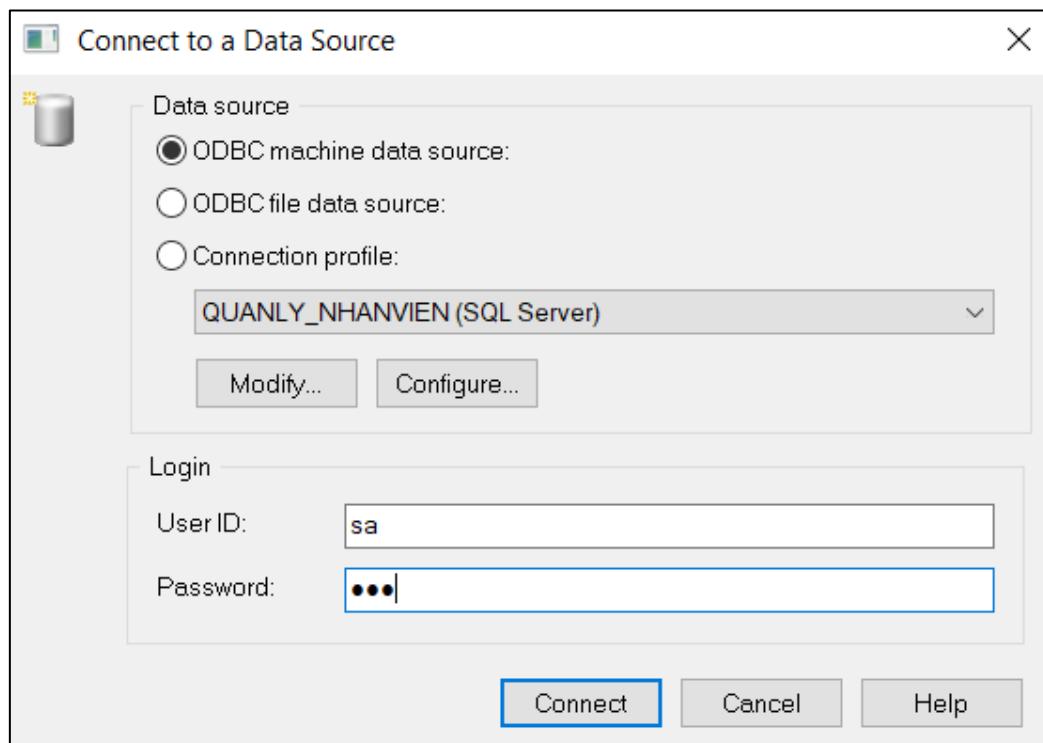
- (11) Kiểm tra kết quả trong mục *Test Results*



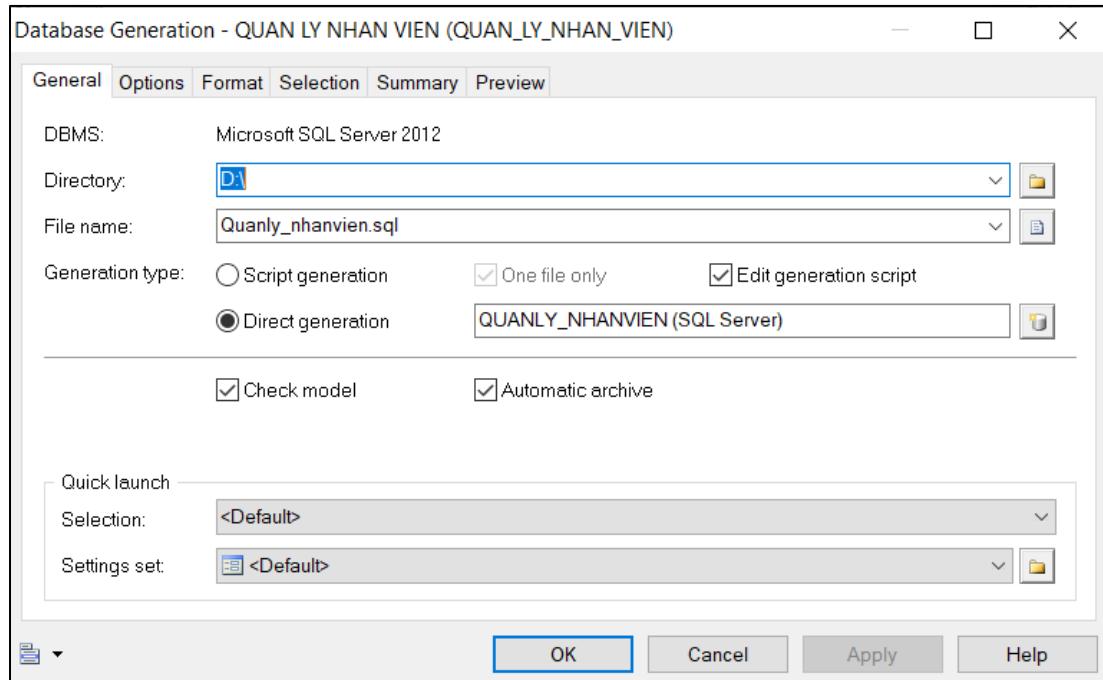
Chọn *OK* để quay về màn hình Connect ban đầu.

- **Bước 3:**

Ở màn hình connect ban đầu, chọn lại kết nối ODBC đã tạo là QUANLY_NHANVIEN trong phần *Machine Data Source*. Nhập thông tin vào ô *User ID* và *Password* để kết nối → chọn *Connect* để kết nối.



Bước 4: chọn **Database** → **Generate Database** (**Ctrl + G**), cửa sổ **Database Generation** xuất hiện, chọn các tham số cần thiết trước khi tạo database:



- Lựa chọn các tham số → chọn **OK**, hộp thoại *Execute SQL Query* hiển thị:

```

alter table PHANCONG
    add constraint FK_PHANCONG_THUC_HIEN_NHANVIEN foreign key (MANV)
        references NHANVIEN (MANV)
go

alter table THANNHAN
    add constraint FK_THANNHAN_CO_NHANVIEN foreign key (MANV)
        references NHANVIEN (MANV)
go

alter table THENHANVIEN
    add constraint FK_THENHANV_CAP_NHANVIEN foreign key (MANV)
        references NHANVIEN (MANV)
go

```

Chọn **Run** để tạo các table Database QUANLY_NHANVIEN trên SQL Server.

1.4. Tạo cơ sở dữ liệu

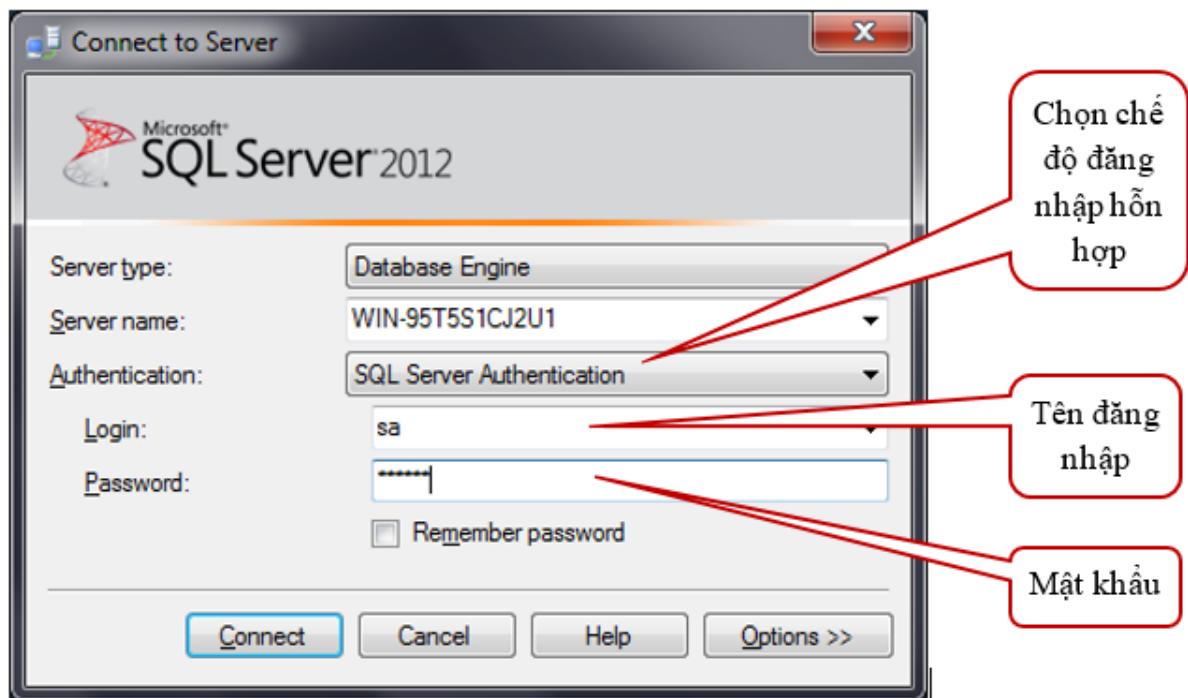
Trong nội dung này, cơ sở dữ liệu QL_NHANVIEN được sử dụng làm cơ sở dữ liệu hướng dẫn thực hành. Tham khảo chi tiết các lược đồ quan hệ trong cơ sở dữ liệu QL_NHANVIEN (Chương 1. Phần 2. Bài tập 3).

Phần hướng dẫn thực hành chương này chỉ sử dụng hai lược đồ quan hệ sau đây trong cơ sở dữ liệu QL_NHANVIEN để làm ví dụ minh họa:

PHONGBAN(MAPH, TENPH, DIADIEM)

NHANVIEN(MANV, HOTEN, NGAYSINH, PHAI, DIACHI, LUONG, MANQL, MAPH)

- ❖ Khởi động hệ quản trị CSDL SQL Server 2012: Vào *Start* → *Programs* → *Microsoft SQL Server 2012* → *SQL Server Management Studio* sẽ xuất hiện cửa sổ đăng nhập như sau:

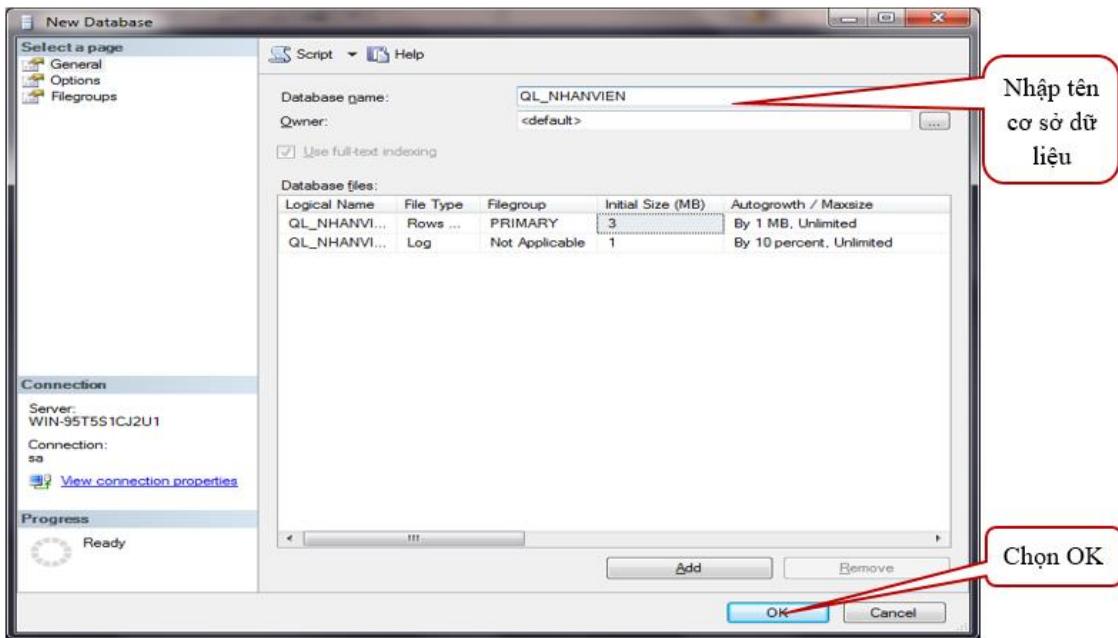


Hình 1.1. Màn hình đăng nhập SQL Server 2012

Sau khi điền đầy đủ các thông tin đăng nhập, nhấn *Connect* để đăng nhập vào hệ thống.

- ❖ Tạo cơ sở dữ liệu

Mở rộng danh mục Server → nhấn chuột phải vào mục *Databases* → chọn *New Database* sẽ xuất hiện màn hình tạo CSDL như sau:



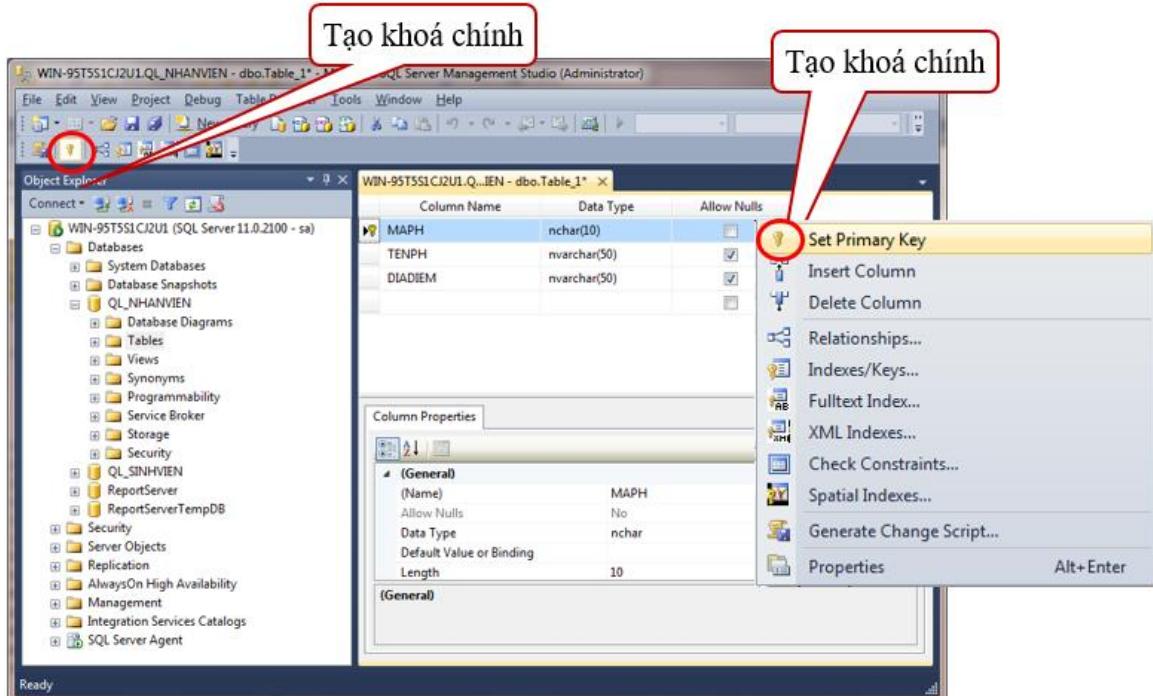
Hình 1.2. Màn hình tạo cơ sở dữ liệu

1.5. Tạo bảng

Trong SQL Server có hai cách tạo bảng: sử dụng công cụ hoặc sử dụng lệnh T-SQL. Với cơ sở dữ liệu QL_NHANVIEN đã tạo ở trên, để tạo bảng PHONGBAN thực hiện như sau:

❖ Sử dụng công cụ trên SQL Server Management Studio

Mở rộng danh mục Database QL_NHANVIEN → nhấp chuột phải vào mục Tables → chọn New Table... xuất hiện màn hình sau:



Hình 1.3. Màn hình tạo khoá chính

Thiết lập thông tin trong cửa sổ tạo bảng như sau:

Column Name (tên cột): Nhập thông tin cột của bảng cần tạo vào các vị trí.

Data Type (kiểu dữ liệu): Chọn kiểu dữ liệu.

Allow Nulls (cho phép null hay không): Đánh dấu nếu thiết lập NULL.

- Tạo khóa chính: Nhấn chuột phải vào cột cần định nghĩa khóa chính, chọn Set Primary Key hoặc chọn biểu tượng khóa () trên thanh công cụ.
- Lưu bảng bằng cách nhấn biểu tượng Save () trên thanh công cụ.

Thực hiện tương tự để tạo bảng NHANVIEN.

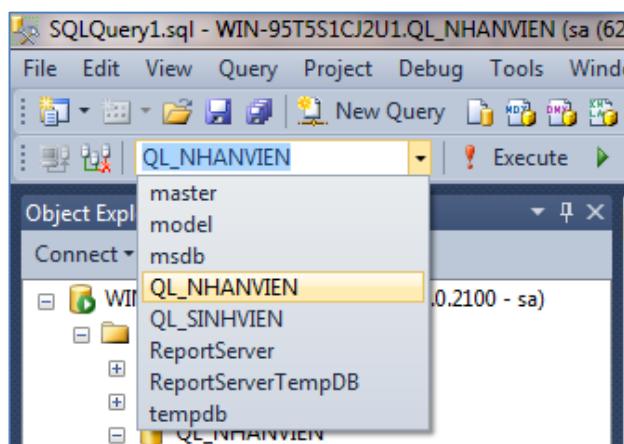
Lưu ý: Bảng được tạo xong hoặc chỉnh sửa xong thì phải đóng cửa sổ tạo cấu trúc bảng lại để hoàn tất thao tác và đảm bảo cho các thao tác khác sẽ thực hiện trên bảng, chẳng hạn như truy vấn hay nhập liệu.

❖ Sử dụng lệnh T – SQL trên Trình soạn thảo lệnh

Chuẩn bị màn hình soạn thảo lệnh: Mở cửa sổ soạn thảo lệnh bằng cách chọn biểu tượng  trên thanh công cụ sẽ xuất hiện màn hình soạn thảo lệnh.

- ✓ Chọn cơ sở dữ liệu muốn tạo bảng.

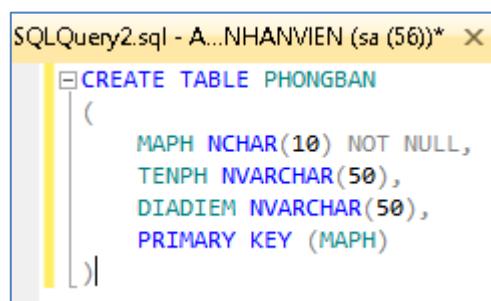
- **Cách 1:** Sử dụng thanh công cụ:



- **Cách 2:** Sử dụng T-SQL

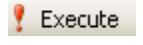
```
USE QL_NHANVIEN  
GO
```

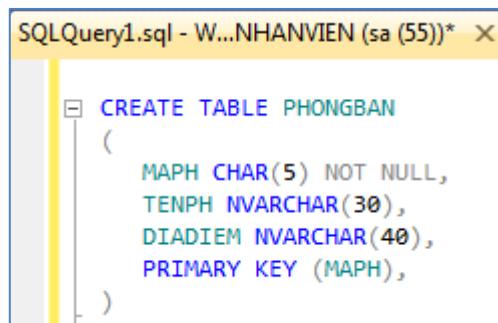
- ✓ Gõ lệnh tạo bảng PHONGBAN vào cửa sổ soạn thảo lệnh:



```
CREATE TABLE PHONGBAN  
(  
    MAPH NCHAR(10) NOT NULL,  
    TENPH NVARCHAR(50),  
    DIADIEM NVARCHAR(50),  
    PRIMARY KEY (MAPH)
```

- ✓ Kiểm tra lỗi cú pháp và thực thi câu lệnh:

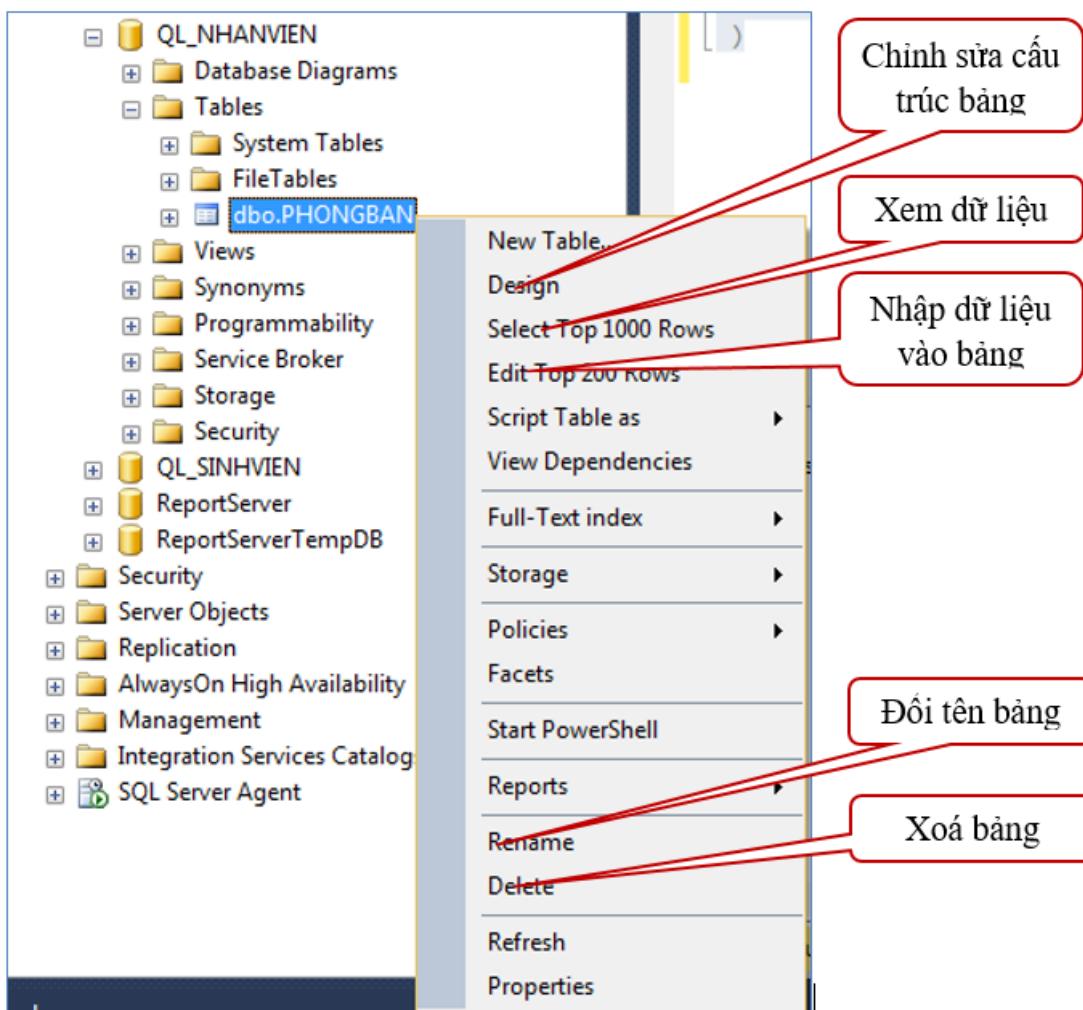
Chọn khái lệnh sau đó kiểm tra lỗi bằng cách nhấn vào biểu tượng:  , nếu không có lỗi xảy ra thì nhấn chuột vào biểu tượng  để thực thi lệnh. (Có thể thực thi lệnh mà không cần kiểm tra lỗi)



```
SQLQuery1.sql - W...NHANVIEN (sa (55))*
CREATE TABLE PHONGBAN
(
    MAPH CHAR(5) NOT NULL,
    TENPH NVARCHAR(30),
    DIADIEM NVARCHAR(40),
    PRIMARY KEY (MAPH),
)
```

Để tạo tiếp bảng NHANVIEN gõ tiếp lệnh tạo bảng NHANVIEN bên dưới lệnh tạo bảng PHONGBAN trước đó rồi chọn khái lệnh vừa gõ, kiểm tra lỗi cú pháp và thực thi tương tự.

Sau khi tạo xong các bảng, có thể thực hiện các thao tác khác như: xem dữ liệu, nhập dữ liệu,...



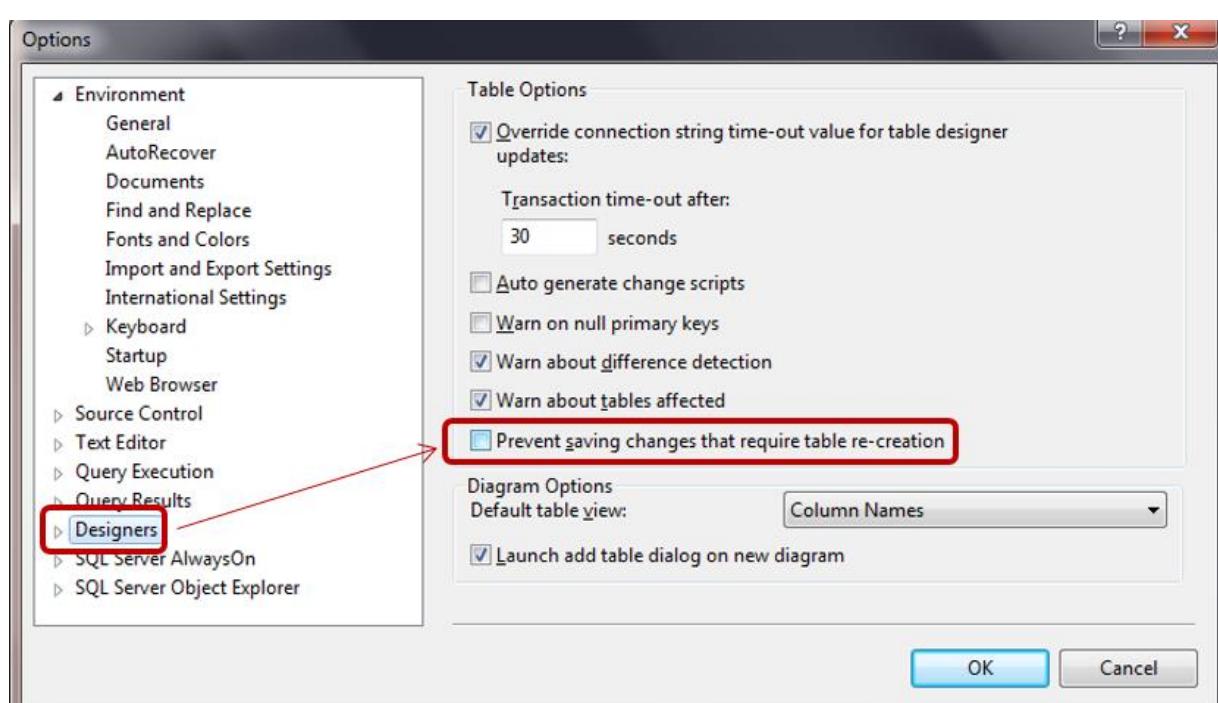
Hình 1.4. Màn hình lựa chọn các thao tác trên bảng

Bên cạnh việc thao tác trực tiếp như trên chúng ta có thể dùng lệnh để thực hiện các hành động này [2].

Lưu ý:

- ✓ Hành động xoá bảng chỉ thực hiện thành công khi không có bất kỳ bảng nào tham chiếu đến nó. Ví dụ: Hai bảng PHONGBAN và NHANVIEN, trên bảng NHANVIEN có thuộc tính khoá ngoại là MAPH tham chiếu đến MAPH trên bảng PHONGBAN, vì vậy không thể xoá bảng PHONGBAN trước. Muốn xoá bảng PHONGBAN thì phải xoá bảng NHANVIEN trước hoặc xoá liên kết khoá ngoại trên bảng NHANVIEN trước khi xoá bảng PHONGBAN.
- ✓ Khi chỉnh sửa cấu trúc bảng, để lưu thông tin chỉnh sửa trên bảng cần phải thực hiện huỷ bỏ chức năng yêu cầu tạo lại bảng như sau:

Chọn *Tools* → *Options* → Xuất hiện cửa sổ *Options*:



1.6. Ràng buộc khoá chính, khoá ngoại

❖ Ràng buộc khoá chính

Khoá chính (Primary key) là một hay tập hợp các thuộc tính dùng để nhận biết duy nhất một bộ trong quan hệ. Khoá chính có thể tạo ra khi tạo bảng hoặc sau đó. Để tạo khoá chính sau khi đã tạo bảng dùng cú pháp sau:

```
ALTER TABLE <tên_bảng>
ADD CONSTRAINT <tên_khoá_chính> PRIMARY KEY(<tên_cột_làm_khoá_chính>)
```

Ví dụ 1.4: Tạo khoá chính cho bảng PHONGBAN như sau:

```
ALTER TABLE PHONGBAN
ADD CONSTRAINT PK_PHONGBAN PRIMARY KEY(MAPH)
```

Trong đó: PK_PHONGBAN là tên khoá chính trên bảng PHONGBAN, PRIMARY KEY (MAPH) là chỉ định thuộc tính MAPH làm khoá chính.

Lưu ý:

- ✓ Phải bảo đảm rằng bảng PHONGBAN đã được tạo trước đó với tình trạng không có khoá chính.
- ✓ PK_PHONGBAN là tên khoá chính nên được đặt theo qui ước hai ký tự đầu là PK (viết tắt của Primary Key) để hiểu là khoá chính, phần tiếp theo của tên là tùy ý nhưng phải mang tính gợi nhớ.
- ✓ Dòng lệnh tạo khoá trên được gõ trong cửa sổ soạn thảo lệnh và phải được thực thi thì khoá chính mới được tạo.

❖ Ràng buộc khoá ngoại

Khoá ngoại (Foreign key) là một hay tập hợp các thuộc tính tham chiếu đến một hay tập hợp các thuộc tính khác (có thể trên cùng bảng hay khác bảng). Khoá ngoại thể hiện mối liên kết 1 – 1 hay 1 – n giữa hai bảng. Có thể tạo khoá ngoại bằng cách dùng chuột kéo thả trên lược đồ Diagram, hoặc có thể sử dụng T – SQL theo cú pháp sau:

```
ALTER TABLE <Tên_bảng>
ADD CONSTRAINT <Tên_khoá_ngoại>
FOREIGN KEY(<Tên_cột_là_khoá_ngoại>)
REFERENCES <Tên_bảng_khoá_ngoại_tham_chiếu_tới>(<Tên_cột_khoá_chính>)
```

Ví dụ 1.5: Tạo khoá ngoại có tên là FK_NV_PB trên bảng NHANVIEN tham chiếu đến MAPH trên bảng PHONGBAN.

```
ALTER TABLE NHANVIEN
ADD CONSTRAINT FK_NV_PB FOREIGN KEY(MAPH)
REFERENCES PHONGBAN(MAPH)
```

Lưu ý:

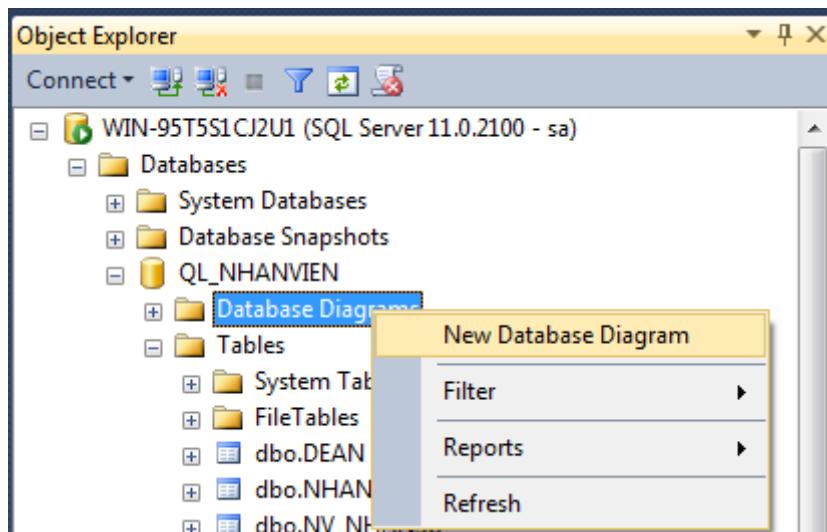
- ✓ Tên khoá ngoại FK_NV_PB nên đặt theo qui ước hai ký tự đầu là FK (viết tắt của FOREIGN KEY), phần còn lại tùy ý nhưng phải gợi nhớ.
- ✓ Phải thực thi lệnh thì khoá ngoại mới được thiết lập.

1.7. Tạo lược đồ Diagram

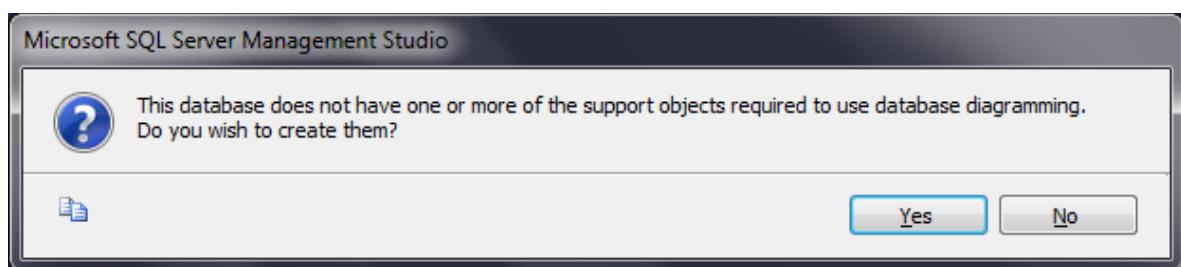
Đối tượng *Diagram* là một lược đồ thể hiện sự liên kết các bảng trong cơ sở dữ liệu với nhau. Trên lược đồ *Diagram* chỉ thể hiện hai loại mối liên kết là 1 – n và 1 – 1. Mỗi mối liên kết tương đương với khoá ngoại trên bảng.

Các bước tạo lược đồ Diagram:

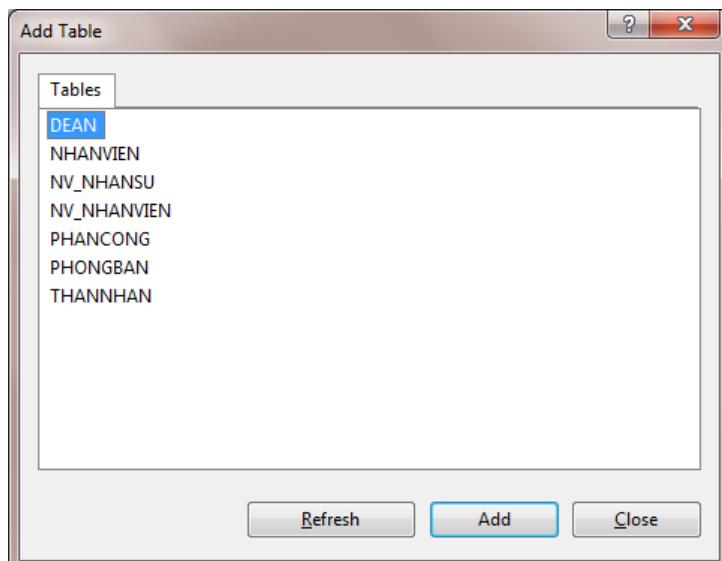
Bước 1: Mở rộng danh mục cơ sở dữ liệu QL_NHANVIEN → nhấn chuột phải vào *Database Diagrams* → chọn *New Database Diagram*.



Nếu xuất hiện thông báo như hình → chọn Yes.



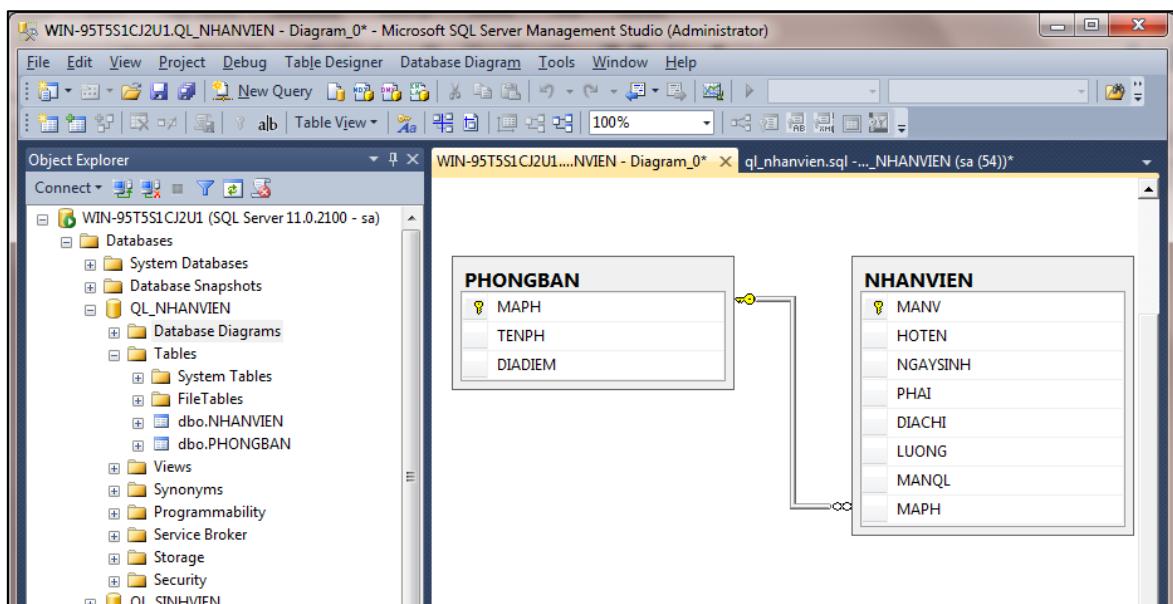
Xuất hiện cửa sổ Add Table.



Trong cửa sổ Add Table chọn các bảng cần đưa vào lược đồ → nhấn nút Add → Close để đóng cửa sổ này lại.

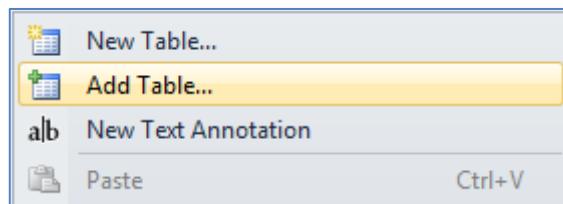
Bước 2: Tạo mối liên kết (khoá ngoại)

Tạo mối liên kết bằng cách nhấp giữ chuột trái vào cột MAPH trên bảng NHANVIEN sau đó kéo sang bảng PHONGBAN và thả chuột → chọn OK, khi đó một mối liên kết 1 – n sẽ được tạo ra.

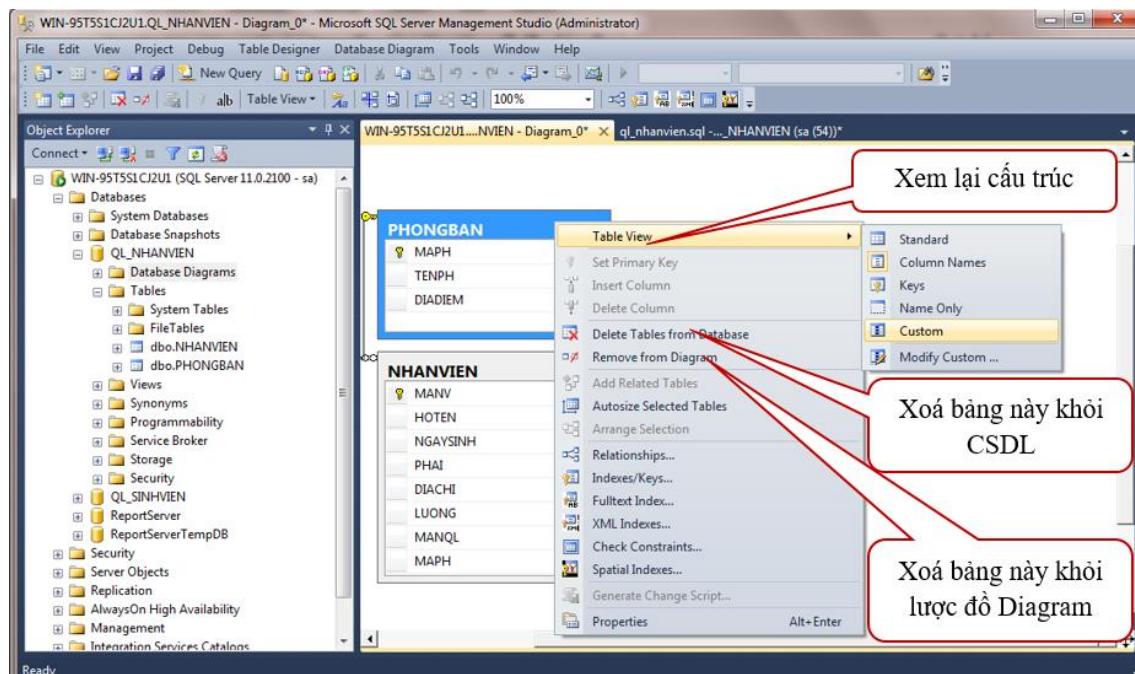


Hình 1.5. Màn hình tạo liên kết trong lược đồ Diagram

Nếu cần bổ sung thêm bảng vào lược đồ thì nhấn chuột phải vào vùng trống → chọn Add Table để mở cửa sổ Add Table như hình sau:



Để chỉnh sửa cấu trúc bảng ngay trên lược đồ Diagram nhấn chuột phải vào bảng → chọn chức năng tương ứng.



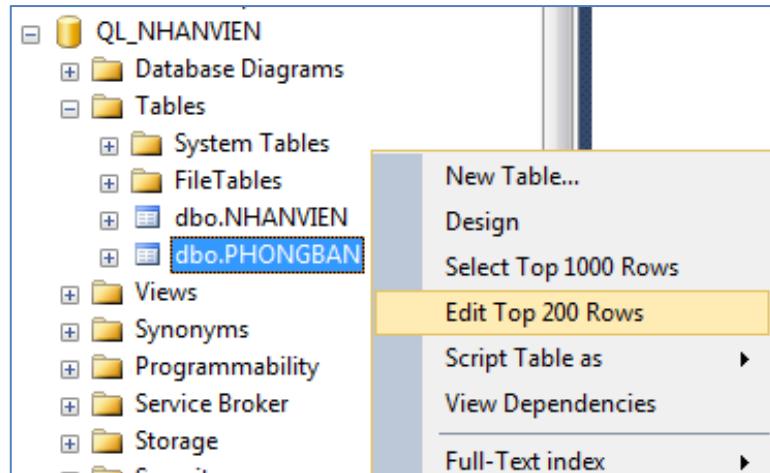
Hình 1.6. Màn hình chọn chức năng trên bảng trong lược đồ Diagram

1.8. Nhập dữ liệu vào bảng

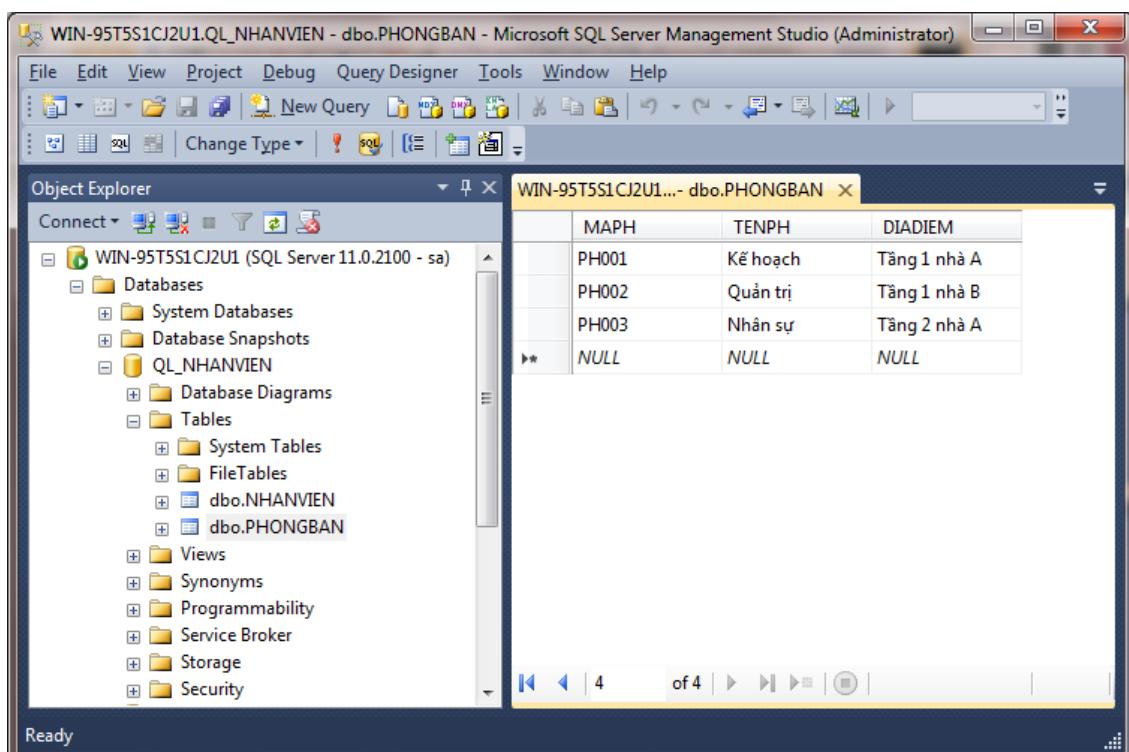
Khi nhập dữ liệu vào bảng phải nhập theo đúng thứ tự, bảng có liên kết một nháp trước, bảng có liên kết nhiều nháp sau. Nếu hai bảng có liên kết 1 – 1 thì nhập dữ liệu trên bảng chứa khoá chính trước, bảng chứa khoá ngoại nhập sau. Sau khi tạo bảng và lược đồ Diagram, tiến hành nhập dữ liệu vào bảng bằng một trong các cách sau:

❖ Sử dụng công cụ trên SQL Server Management Studio

Trong cơ sở dữ liệu QL_NHANVIEN mở rộng danh mục *Tables* → nhấn chuột phải vào bảng PHONGBAN → chọn *Edit Top 200 Rows*.



Hình 1.7. Màn hình chọn chức năng nhập dữ liệu trực tiếp trên bảng
→ Xuất hiện cửa sổ nhập liệu:



Hình 1.8. Màn hình nhập dữ liệu trực tiếp trên bảng

Tuy nhiên, việc nhập dữ liệu vào bảng trực tiếp bằng công cụ không hỗ trợ người sử dụng trong trường hợp thao tác trên các trình ứng dụng khác có kết nối với SQL Server. Để giải quyết vấn đề này có thể nhập dữ liệu bằng cách sử dụng lệnh T – SQL.

❖ Sử dụng lệnh T – SQL

Cú pháp:

```
INSERT INTO <Tên_bảng> [Tên_các_cột]
VALUES (các giá trị tương ứng)
```

Ví dụ 1.1: Nhập dữ liệu vào bảng PHONGBAN trên cửa sổ soạn thảo lệnh.

The screenshot shows a SQL query window with the following content:

```
INSERT INTO PHONGBAN
VALUES ('PH005', N'Đầu tư', N'Tầng 2 Nhà B');

(1 row(s) affected)
```

The window includes a status bar showing "100 %". Below the query window is a "Messages" pane which is currently empty.

Sau khi gõ xong lệnh, chọn khối lệnh vừa gõ rồi kiểm tra lỗi cú pháp và thực thi. Nếu không có lỗi xảy ra thì một dòng dữ liệu được thêm vào bảng PHONGBAN.

Lưu ý:

- ✓ Đối với các cột có kiểu dữ liệu chuỗi (char, varchar, nchar, nvarchar) và ngày tháng (date, datetime, smalldatetime,...) thì giá trị nhập vào phải đặt trong cặp dấu nháy đơn “ ”.
- ✓ Giá trị nhập vào cho các cột kiểu số thì không đặt trong cặp dấu nháy đơn “ ”.
- ✓ Nếu ngày tháng trong hệ thống có định dạng khác với giá trị ngày tháng cần nhập thì dùng lệnh **SET DATEFORMAT <định dạng>** để định dạng lại ngày tháng trên SQL Server. Ví dụ: **SET DATEFORMAT DMY**, trong đó: DMY là các tham số Date, Month, Year. Các tham số này có thể thay đổi theo yêu cầu định dạng của người dùng.

Nếu muốn nhập nhiều dòng dữ liệu cùng lúc thì dùng cú pháp sau:

```
INSERT INTO <Tên_bảng> [Tên_các_cột]
VALUES    (các giá trị tương ứng 1),
          (các giá trị tương ứng 2),
```

Ví dụ 1.2: Viết lệnh nhập thông tin nhân viên với ngày sinh được thiết lập theo định dạng DMY.

```

SET DATEFORMAT DMY
INSERT INTO NHANVIEN(MANV, HOTEN, NGAYSINH)
VALUES('NV0001', 'N'Nguyễn Văn Nam', '12/07/1988')

```

	MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
1	NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL

1.9. Tạo bảng ảo

Bảng ảo (View) là một đối tượng được tạo ra từ các bảng cơ sở. Mục đích của việc tạo bảng ảo là để tạo ra các bảng trung gian nhằm đơn giản hóa các câu truy vấn phức tạp, che dấu dữ liệu, bảo mật dữ liệu. Có thể sử dụng công cụ để tạo bảng ảo như sau:

Từ màn hình quản trị *SQL Server Management Studio*, mở rộng danh mục cơ sở dữ liệu QL_NHANVIEN > nhấp chuột phải vào mục Views > chọn New View > xuất hiện cửa sổ Add Table.

Trên cửa sổ Add Table chọn các bảng liên quan đến bảng ảo cần tạo, sau đó chọn các cột thuộc tính:

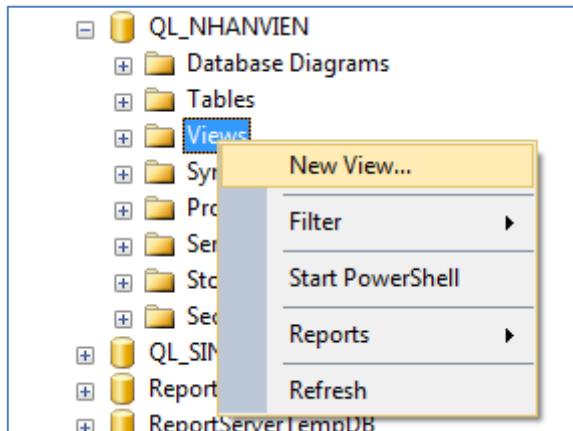
- ✓ **Column:** Tên các cột cần hiển thị.
- ✓ **Alias:** Tên bí danh
- ✓ **Filter:** Điều kiện lọc

Sau khi hoàn tất thì chọn biểu tượng Save để lưu lại.

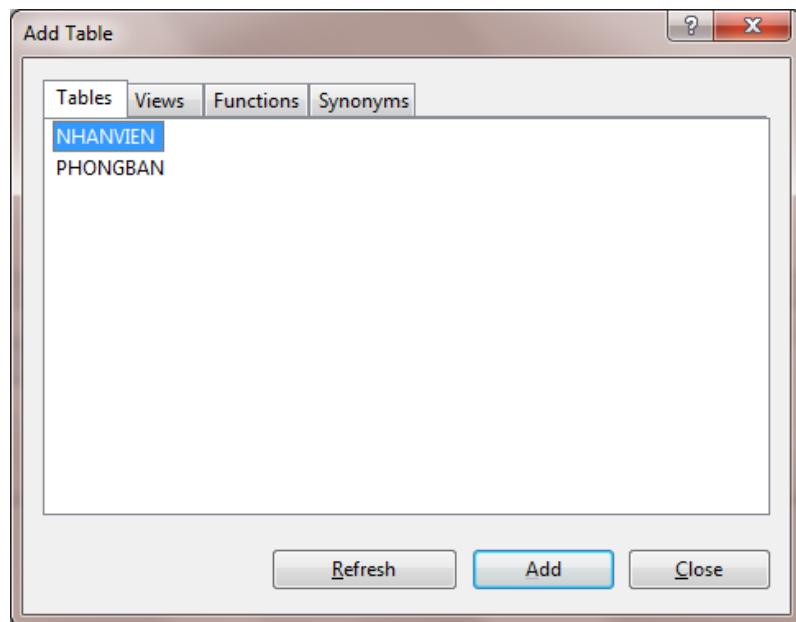
Ví dụ 1.3: Cần tạo một bảng ảo lưu trữ thông tin nhân viên nữ gồm 3 cột: MANV, HOTEN, TENPH có tên là NV_NU lấy từ hai bảng PHONGBAN, NHANVIEN.

Các bước thực hiện:

- **Bước 1:** Nhấp chuột phải vào mục View → chọn New View.



Trên cửa sổ *Add Table* chọn các bảng PHONGBAN, NHANVIEN → chọn *Add* → chọn *Close* để đóng cửa sổ này lại.



Hình 1.9. Màn hình chọn bảng tạo bảng ảo

- **Bước 2:** chọn cột MANV, HOTEN, TENPH và gõ điều kiện tại vị trí *Filter* của cột PHAI như trong hình.

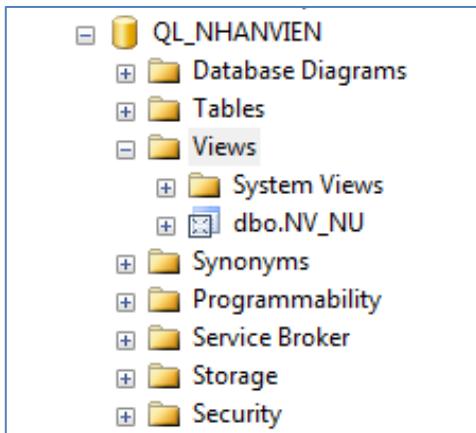
Column	Alias	Table	Sort Type	Sort Order	Filter
MANV		NHANVIEN			
HOTEN		NHANVIEN			
TENPH		PHONGBAN			
PHAI		NHANVIEN			Nữ

```

SELECT      dbo.NHANVIEN.MANV, dbo.NHANVIEN.HOTEN, dbo.PHONGBAN.TENPH
FROM        dbo.NHANVIEN INNER JOIN
                      dbo.PHONGBAN ON dbo.NHANVIEN.MAPH = dbo.PHONGBAN.MAPH
WHERE       (dbo.NHANVIEN.PHAI = N'Nữ')
  
```

Hình 1.10. Màn hình tạo bảng ảo

- **Bước 3:** Sau khi thiết lập xong bảng ảo, chọn *Save* để lưu lại và đóng cửa sổ tạo bảng ảo. Bảng ảo sau khi tạo thành công sẽ xuất hiện trong mục *Views* như hình vẽ. Khi đó có thể mở, chỉnh sửa hay xoá như cách làm đối với bảng.



PHẦN 2: BÀI TẬP THỰC HÀNH

Bài 1: Hệ thống Quản lý sinh viên trong một Trường Đại học được khảo sát và mô tả như sau:

- Trường Đại học có nhiều khoa, mỗi khoa có một mã số duy nhất, có tên khoa.
- Mỗi giảng viên có một mã số duy nhất, có họ tên tên và thuộc một khoa duy nhất.
- Mỗi lớp học có một mã lớp duy nhất, có tên lớp, số lượng của lớp, thông tin lớp trưởng, mỗi lớp thuộc một khoa duy nhất.
- Mỗi sinh viên có một mã số duy nhất, có họ tên, thuộc một phái, có một ngày sinh, một địa chỉ và theo học một lớp duy nhất.
- Mỗi môn học có một mã môn học duy nhất, tên môn học, số tín chỉ tương ứng. Đối với mỗi môn học có nhiều sinh viên học, có nhiều giảng viên giảng dạy.
 - Khi sinh viên thi xong, cần ghi nhận lại điểm số mỗi môn học của sinh viên, mỗi môn học sinh viên có thể thi nhiều lần, thông tin bảng điểm gồm: Mã môn học, mã sinh viên, lần thi và điểm thi.
 - Trong mỗi học kỳ của từng năm học, các hoạt động giảng dạy diễn ra với mỗi lớp học phần do một giảng viên đảm nhận, thông tin hoạt động giảng dạy gồm: Mã môn học, học kỳ, năm học, giảng viên giảng dạy.
- Mỗi sinh viên có một hoặc nhiều thân nhân để liên hệ. Mỗi thân nhân cần lưu trữ mã thân nhân, họ tên thân nhân, giới tính, mối quan hệ giữa sinh viên và thân nhân.

Yêu cầu:

- Xác định các thực thể.
- Xác định thuộc tính của các thực thể.

- c. Xác định mối kết hợp.
- d. Xây dựng mô hình thực thể kết hợp cho hệ thống quản lý sinh viên.
- e. Sử dụng phần mềm (tùy chọn) để vẽ mô hình thực thể kết hợp.
- f. Chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ.
- g. Xác định khóa chính, khóa ngoại cho từng quan hệ.

Bài 2: Một hệ thống quản lý bán hàng được mô tả như sau:

Một cửa hàng chuyên bán sỉ và lẻ các mặt hàng đủ loại. Người quản lý cửa hàng cần xây dựng một ứng dụng quản lý công việc đặt hàng, giao hàng và bán hàng tại cửa hàng. Sau đây là kết quả của việc phân tích yêu cầu ứng dụng:

- Cửa hàng bán ra trên 300 mặt hàng với nguồn hàng lấy từ các nhà cung cấp. Một nhà cung cấp có một mã nhà cung cấp, tên, địa chỉ và điện thoại của nhà cung cấp. Mỗi nhà cung cấp có thể cung ứng nhiều mặt hàng khác nhau và mỗi mặt hàng cũng có thể được cung cấp bởi nhiều nhà cung cấp khác nhau, cần ghi nhận lại nhà cung cấp nào có thể cung ứng những mặt hàng gì.
- Cần lưu lại thông tin về tất cả các mặt hàng mà cửa hàng có mua bán: Mã mặt hàng, tên hàng, hàng thuộc loại nào, đơn vị tính, quy cách, số lượng tồn. Mỗi loại có mã loại hàng, tên loại hàng.
- Mỗi lần đặt hàng, cửa hàng sẽ điền một phiếu đặt hàng gửi đến nhà cung cấp. Cửa hàng phải điền các thông tin sau vào đơn đặt hàng: số đơn đặt hàng, ngày đặt hàng là bao nhiêu. Cuối mỗi đơn đặt hàng có thông tin về tổng số mặt hàng cần đặt. Trên phiếu đặt hàng chỉ có các mặt hàng mà nhà cung cấp có thể cung ứng.
- Mỗi khi đến giao hàng, nhà cung cấp giao cho cửa hàng một phiếu giao hàng, gồm các thông tin sau: số phiếu giao, ngày giao, giao cho đơn đặt hàng nào, mỗi mặt hàng số lượng giao và đơn giá là bao nhiêu. Ứng với 1 lần đặt hàng, nhà cung cấp có thể giao hàng tối đa 3 lần và không được trễ hơn 7 ngày so với ngày đặt. Nhà cung cấp chỉ được giao cá mặt hàng mà nhà cung cấp đó đặt với số lượng giao không lớn hơn số lượng đặt.
- Khi khách hàng đến mua hàng, cửa hàng sẽ lưu lại thông tin tất cả các đơn bán hàng để tiện việc kiểm hàng. Thông tin hóa đơn gồm: số hóa đơn, ngày lập hóa đơn, tên khách hàng, địa chỉ khách hàng, điện thoại khách hàng, khách hàng đã mua những mặt hàng nào với số lượng mua, đơn giá mua là bao nhiêu.

Yêu cầu:

- a. Xác định các thực thể.
- b. Xác định thuộc tính của các thực thể.
- c. Xác định mối kết hợp.

- d. Xây dựng mô hình thực thể kết hợp cho hệ thống quản lý bán hàng.
- e. Sử dụng phần mềm (tùy chọn) để vẽ mô hình thực thể kết hợp.
- f. Chuyển đổi từ mô hình thực thể kết hợp sang mô hình quan hệ.
- g. Xác định khóa chính, khóa ngoại cho từng quan hệ.

Bài 3: Cho lược đồ cơ sở dữ liệu quản lý nhân viên gồm có các lược đồ quan hệ sau:

PHONGBAN(MAPH, TENPH, DIADIEM)

NHANVIEN (MANV, HOTEN, NGAYSINH, PHAI, DIACHI, LUONG, MANQL, MAPH)

DEAN (MADA, TENDA, DIADIEMDA, NGAYBD)

PHANCONG (MANV, MADA, SONGAYTHAMGIA)

THANNHAN (MANV, TENTN, PHAI, NGAYSINH, QUANHE)

THENHANVIEN(MATHE,MANV, NGAYCAP)

Yêu cầu:

- a. Khởi động SQL Server Management Studio, đăng nhập và tạo một cơ sở dữ liệu có tên là QL_NHANVIEN.
- b. Trong cơ sở dữ liệu QL_NHANVIEN tạo các bảng như hình dưới đây.

Lưu ý: *Tạo khoá chính trong khi tạo bảng.*

Bảng PHONGBAN

Column Name	Data Type	Allow Nulls
MAPH	char(5)	<input type="checkbox"/>
TENPH	nvarchar(30)	<input type="checkbox"/>
DIADIEM	nvarchar(40)	<input checked="" type="checkbox"/>

Bảng NHANVIEN

Column Name	Data Type	Allow Nulls
MANV	char(6)	<input type="checkbox"/>
HOTEN	nvarchar(40)	<input checked="" type="checkbox"/>
NGAYSINH	date	<input checked="" type="checkbox"/>
PHAI	nchar(3)	<input checked="" type="checkbox"/>
DIACHI	nvarchar(40)	<input checked="" type="checkbox"/>
LUONG	money	<input checked="" type="checkbox"/>
MANQL	char(6)	<input checked="" type="checkbox"/>
MAPH	char(5)	<input checked="" type="checkbox"/>

Bảng DEAN

Column Name	Data Type	Allow Nulls
MADA	char(5)	<input type="checkbox"/>
TENDA	nvarchar(50)	<input checked="" type="checkbox"/>
DIADIEMDA	nvarchar(30)	<input checked="" type="checkbox"/>
NGAYBD	date	<input checked="" type="checkbox"/>

Bảng PHANCONG

Column Name	Data Type	Allow Nulls
MANV	char(6)	<input type="checkbox"/>
MADA	char(5)	<input type="checkbox"/>
SONGAYTG	int	<input checked="" type="checkbox"/>

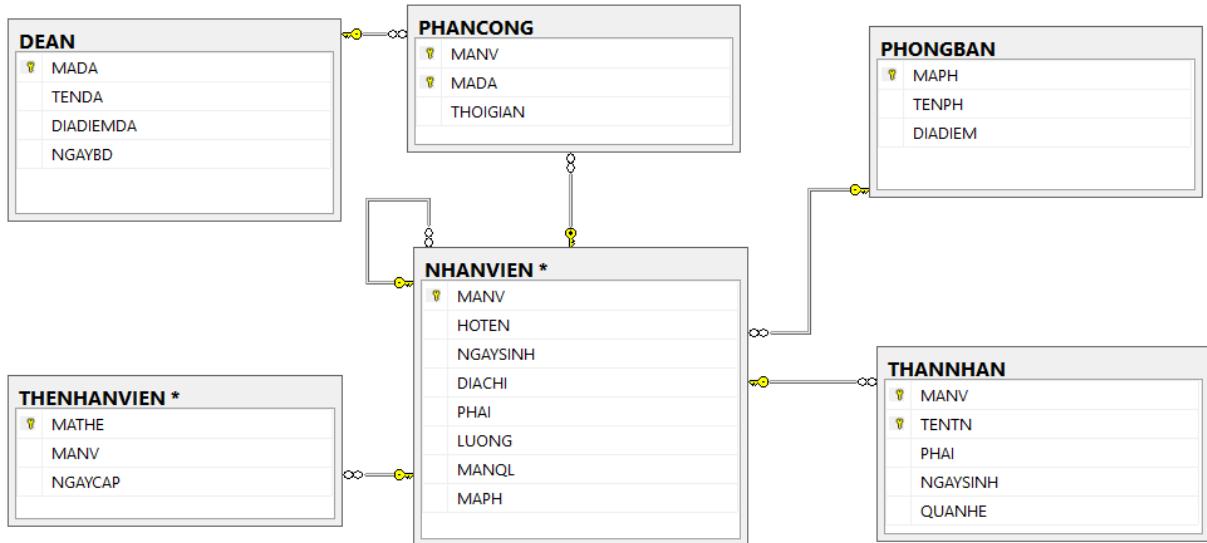
Bảng THANNHAN

Column Name	Data Type	Allow Nulls
MANV	char(6)	<input type="checkbox"/>
TENTN	nvarchar(40)	<input type="checkbox"/>
PHAI	nchar(3)	<input checked="" type="checkbox"/>
NGAYSINH	date	<input checked="" type="checkbox"/>
QUANHE	nvarchar(15)	<input checked="" type="checkbox"/>

Bảng THENHANVIEN

Column Na...	Data Type	Allow Nulls
MATHE	char(10)	<input type="checkbox"/>
MANV	char(6)	<input type="checkbox"/>
NGAYCAP	datetime	<input checked="" type="checkbox"/>

c. Tạo lược đồ Diagram liên kết các bảng trên với nhau như hình sau:



- d. Từ màn hình Diagram thực hiện chỉnh sửa kiểu dữ liệu của cột NGAYSINH trên bảng NHANVIEN, NGAYSINH trên bảng THANNHAN thành SmallDatetime.
(Hướng dẫn: Từ màn hình Diagram nhấn chuột phải vào bảng cần chỉnh sửa > chọn Table View > chọn Custom > chỉnh sửa theo yêu cầu.)
- e. Từ màn hình Diagram thực hiện xoá cột NGAYSINH trên bảng THANNHAN và thêm cột TENNV trên bảng THANNHAN.
- f. Từ danh mục Table thực hiện thêm cột TENNQL trên bảng PHONGBAN và xóa cột NGAYBD trên bảng DEAN.
(Hướng dẫn: Nhấn chuột phải vào bảng cần chỉnh sửa > chọn Design > Chọn cột cần xóa > Chọn Delete Column).
- g. Nhập dữ liệu vào các bảng như dưới đây:

PHONGBAN

MAPH	TENPH	DIADIEM
PH001	Kế hoạch	Tầng 1 nhà A
PH002	Quản trị	Tầng 1 nhà B
PH003	Nhân sự	Tầng 2 nhà A
PH004	Tài vụ	Tầng 3 nhà A
PH005	Đầu tư	Tầng 2 nhà B
PH006	Vật tư	Tầng 3 nhà B
PH007	Tư vấn	Tầng 3 nhà B

NHANVIEN

MANV	HOTENNV	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
NV0001	Nguyễn Văn Nam	12-07-1988	Nam	Tây Ninh	15000000	NV0009	PH003
NV0002	Nguyễn Kim Ánh	10-02-1990	Nữ	TP. HCM	8000000	NV0009	PH003
NV0003	Nguyễn Thị Châu	12-10-1979	Nữ	Vũng Tàu	12000000	NV0006	PH003
NV0004	Trần Văn Út	23-08-1977	Nam	Hà Nội	7000000	NV0005	PH002
NV0005	Trần Lệ Quyên	22-12-1987	Nữ	Hà Nội	9000000	NV0005	PH002
NV0006	Bùi Đức Chí	22-12-1987	Nam	TP. HCM	10000000	NV0008	PH003
NV0007	Nguyễn Tuấn Anh	06-09-1991	Nam	Tây Ninh	35000000	NV0002	PH003
NV0008	Đỗ Xuân Thuỷ	14-05-1985	Nam	TP. HCM	21000000	NV0002	PH002
NV0009	Trần Minh Tú	17-09-1985	Nam	Đồng Nai	18000000	NULL	PH001
NV0010	Trần Khánh An	13-11-1987	Nữ	Khánh Hòa	12000000	NULL	NULL
NV0011	Nguyễn Ngọc Phan	02-06-1995	Nam	Đồng Nai	13000000	NULL	NULL

DEAN

MADA	TENDA	DIADIEMDA	NGABD
DA001	Đèn bù giải toả	Phường 12, Q. Tân Bình	01/01/2015
DA002	Giải phóng mặt bằng	Phường 12, Q. Tân Bình	01/06/2015
DA003	Cải tạo mặt đường số 9	Phường Tây Thạnh, Q. Tân Phú	01/01/2016
DA004	Bắt đầu thi công	Phường 26, Q. Bình Thạnh	04/05/2016
DA005	Hoàn thiện mặt bằng	Phường Tân Quý, Quận 7	10/12/2016

PHANCONG

MANV	MADA	THOIGIAN
NV0001	DA001	05/02/2015
NV0001	DA003	17/03/2016
NV0003	DA003	01/01/2016
NV0005	DA004	10/05/2016
NV0007	DA005	20/12/2016

THANHAN

MANV	TENTN	PHAI	NGAYSINH	QUANHE
NV0001	Nguyễn Thị Tám	Nữ	05/09/2015	Con
NV0001	Nguyễn Văn Bình	Nam	22/05/1983	Anh
NV0002	Nguyễn Chính Nghĩa	Nam	07/03/1998	Em
NV0005	Lê Anh Hùng	Nam	05/04/1978	Chồng
NV0006	Bùi Đại An	Nam	03/12/1976	Anh
NV0008	Lê Thảo Nguyên	Nữ	12/06/1985	Vợ
NV0009	Trần Thanh Nhàn	Nữ	30/05/1979	Chị

THENHANVIEN

MATHE	MANV	NGAYCAP
T0001	NV0001	05/03/2018
T0002	NV0002	17/03/2019
T0003	NV0003	01/01/2020
T0004	NV0004	10/05/2020
T0005	NV0005	20/12/2021

Bài 4: Khởi động SQL Server Management Studio, đăng nhập và tạo một cơ sở dữ liệu có tên là QL_LINHKIEN.

- a. Dùng lệnh T – SQL thực hiện tạo các bảng tương ứng với lược đồ cơ sở dữ liệu sau:

LOAILK(MALOAI, TENLOAI)
LINHKIEN(MALK, TENLK, NGAYSX, TGBH, MALOAI, NSX, DVT)
KHACHHANG(MAKH, TENKH, DCHI, DTHOAI)
HOADON(MAHD, NGAYHD, MAKH, TONGTIEN)
CHITIETHD(MAHD, MALK, SOLUONG, DONGIA)

- b. Dùng lệnh T – SQL tạo ràng buộc khoá ngoại có trong cơ sở dữ liệu (tương ứng với các đường mũi tên tham chiếu).
- c. Tạo lược đồ Diagram.
- d. Dùng lệnh INSERT INTO nhập liệu vào các bảng với dữ liệu như dưới đây:

LOAILK

MALOAI	TENLOAI
MOU	Chuột
LAP	Máy tính xách tay
CPU	Bộ xử lý
PCX	Máy tính để bàn
MAI	Mainboard

LINH KIEN

MALK	TENLK	NGAYSX	TGBH (Tháng)	MALOAI	NSX	DVT
MOU001	Chuột quang có dây	01/01/2014	12	MOU	Genius	Cái
MOU002	Chuột quang không dây	04/02/2015	12	MOU	Mitsumi	Cái
MOU003	Chuột không dây	02/04/2014	24	MOU	Abroad	Cái
CPU001	CPU ADM	05/04/2015	24	CPU	Abroad	Cái
CPU002	CPU INTEL	07/02/2016	36	CPU	Mitsumi	Cái
CPU003	CPU ASUS	08/12/2015	36	CPU	Abroad	Cái
MAI001	Mainboard ASUS	04/12/2015	36	MAI	Mitsumi	Cái
MAI002	Mainboard ATXX	03/03/2016	12	MAI	Mitsumi	Cái
MAI003	Mainboard ACER	14/04/2015	12	MAI	Genius	Cái
PCX001	Acer 20144	19/10/2015	12	PCX	Acer	Bộ

KHACHHANG

MAKH	TENKH	DIACHI	DTHOAI
KH001	Nguyễn Thu Tâm	Tây Ninh	0989751723
KH002	Đinh Bảo Lộc	Lâm Đồng	0918234654
KH003	Trần Thanh Diệu	TP. HCM	0978123765
KH004	Hồ Tuấn Thành	Hà Nội	0909456768
KH005	Huỳnh Kim Ánh	Khánh Hòa	0932987567

HOADON

MAHD	NGAYHD	MAKH	TONGTIEN
HD001	01/04/2015	KH001	
HD002	15/05/2016	KH005	
HD003	14/06/2016	KH004	
HD004	03/06/2016	KH005	
HD005	05/06/2016	KH001	
HD006	07/07/2016	KH003	
HD007	12/08/2016	KH002	
HD008	25/09/2016	KH003	

CHITIETHD

MAHD	MALK	SOLUONG	DONGIA
HD001	MOU001	2	1000000
HD002	MOU002	1	2000000
HD003	MOU003	6	3000000
HD004	CPU001	5	500000
HD005	CPU002	6	560000
HD006	CPU003	3	400000
HD006	MAI001	1	200000
HD007	MAI002	1	150000
HD007	MAI003	2	160000
HD007	MOU001	1	1000000
HD008	CPU001	2	500000

PHẦN 3: BÀI TẬP LÀM THÊM

Bài 1: Hãy tạo một cơ sở dữ liệu có tên là **QL_THUVIEN** và nhập liệu theo cấu trúc sau:

TACGIA (MATG, TENTG, DIACHI)

MATG	TENTG	DIACHI
TG001	Nguyễn Hữu Anh	Q3, TP. HCM
TG002	Tô Hoài	Bình Thạnh, TP. HCM
TG003	Nguyễn Quang Sang	Trảng Bàng, Tây Ninh
TG004	Hồ Tùng Mậu	Trảng Bom, Đồng Nai
TG005	Thi Anh Trung	Bến Lức, Long An
TG006	Nguyễn Nhật Ánh	Q1, TP. HCM

LOAISACH (MALOAI, TENLOAI)

MALOAI	TENLOAI
SH	SÁCH
TR	TRUYỆN
TC	TẠP CHÍ

NHAXUATBAN (MANXB, TENNXB, DCNXB, DTNXB)

MANXB	TENNXB	DCNXB	DTNXB
NXB01	Đại học quốc gia TP.HCM	Quận 1, TP. HCM	08.765348758
NXB02	Đại học quốc gia Hà Nội	Quận Cầu Giấy, Hà Nội	04.765476574
NXB03	Thanh Niên	Quận 3, TP. HCM	08.765475665
NXB04	Tổng Hợp	Quận Thanh Xuân, Hà Nội	04.876847676
NXB05	Nhi Đồng	Quận 10, TP. HCM	08.765746767

SACH (MASH, TENSH, MATG, NAMXB, MANXB, MALOAI)

MASH	TENSH	MATG	NAMXB	MANXB	MALOAI
SH001	Toán rời rạc	TG001	1998	NXB01	SH
TR001	Dế mèn phiêu lưu ký	TG002	1997	NXB05	TR
TR002	Bàn có 5 chỗ ngồi	TG006	2000	NXB05	TR
TC001	Tạp chí Tin học và Điều khiển số 6/2015	TG003	2015	NXB04	TC
TC002	Tạp chí Tin học và Điều khiển số 9/2015	TG003	2015	NXB04	TC
SH002	Kỹ thuật lập trình C#	TG004	1998	NXB04	SH

DOCGIA (MADG, TENDG, NGAYSINH, GIOITINH, LIENHE)

MADG	TENDG	NGAYSINH	GIOITINH	LIENHE
DG001	Nguyễn Thanh Nam	13/09/1990	Nam	01681234567
DG002	Lê Văn Anh	24/04/1994	Nam	0982999011
DG003	Trần Thanh Bình	30/10/1989	Nữ	01659000123
DG004	Nguyễn Thị Thanh	02/02/1997	Nữ	01231452370

MUONTRASACH (MADG, MASH, NGAYMUON, NGAYTRA)

MADG	MASH	NGAYMUON	NGAYTRA
DG004	SH002	21/05/2016	28/05/2016
DG001	SH001	01/06/2016	02/06/2016
DG003	TC001	01/06/2016	11/06/2016
DG001	SH002	01/06/2016	NULL

Bài 2: Hãy tạo một cơ sở dữ liệu có tên là QL_THUENHA và nhập liệu theo cấu trúc sau:

LOAINHA (MALN, TENLN, GHICHU)

MALN	TENLN	GHICHU
NC1	Cấp 1	NULL
NC2	Cấp 2	NULL
NC3	Cấp 3	NULL
NC4	Cấp 4	NULL

NHA (MANHA, DIACHI, TIENTHUE, MALN)

MANHA	DIACHI	TIENTHUE	MALN
C2001	142 Lê Trọng Tấn, Tân Phú, TP. HCM	10000000	NC2
C3001	111 Bình Thới, Q11, TP. HCM	7000000	NC3
C4001	123 Tân Thới Nhất, Quận 12, TP. HCM	4000000	NC4
C4002	21 Tô Ký, Quận 12, TP. HCM	3500000	NC4

KHACHHANG (MAKH, TENKH, DIACHI, DTHOAI, KHANANGTHUE)

MAKH	TENKH	DIACHI	DTHOAI	KHANANGTHUE
KH001	Trần Thanh Bình	123, Trương Định, Q3, TP. HCM	0989123456	10000000
KH002	Trần Thị Lan	01, Lê Lai, Q1, TP. HCM	0918456234	9000000
KH003	Lê Thành Nam	12, Ngô Quyền, Q5, TPP. HCM	0909657456	11000000
KH004	Nguyễn Thảo Mi	120, Thành Thái, Q10, TP. HCM	0912657489	3000000
KH005	Lê Anh Tuấn	123, Trương Định, Q3, TP. HCM	0989768567	4000000

DANGTIN (MANHA, NGAYDANG, THOIHANDT)

MANHA	NGAYDANG	THOIHANDT (NGÀY)
C2001	01/05/2016	7
C4002	02/05/2016	10
C4001	04/06/2016	30
C4001	13/06/2016	7
C3001	12/06/2016	7

XEMNHA (MANHA, MAKH, NGAYXEMNHA, KETLUAN)

MANHA	MAKH	NGAYXEMNHA	KETLUAN
C2001	KH003	07/05/2016	NULL
C4002	KH002	07/05/2016	NULL
C4001	KH004	05/06/2016	NULL
C4001	KH002	14/06/2016	NULL
C3001	KH001	15/06/2016	NULL

Bài 3: Hãy tạo một cơ sở dữ liệu có tên là QL_CHUNGCHI và nhập liệu theo cấu trúc sau:

LOAICC (MALCC, TENLCC)

MALCC	TENLCC
CCA	Chứng chỉ A
CCB	Chứng chỉ B
CAD	Chứng chỉ AutoCad

CHUNGCHI (MACC, TENCC, THOIGIANHOC, MALCC)

MACC	TENCC	THOIGIANHOC (tuần)	MALCC
CCA1	Chứng chỉ A hạng 1	5	CCA
CCA2	Chứng chỉ A hạng 2	6	CCA
CCB1	Chứng chỉ B hạng 1	6	CCB
CCB2	Chứng chỉ B hạng 2	8	CCB
CAD1	Chứng chỉ AutoCad	9	CAD

HOCVIEN (MAHV, HOTEN, NGAYSINH, GIOITINH, DIACHI)

MAHV	HOTEN	NGAYSINH	GIOITINH	DIACHI
HV001	Trần Thanh Bình	4/12/1998	Nữ	123, Trương Định, Q3, TP. HCM.
HV002	Trần Thị Lan	12/1/1997	Nữ	1, Paster, Q3, TP. HCM
HV003	Lê Thành Nam	23/6/1999	Nam	12, Ngô Quyền, Q5, TP. HCM
HV004	Nguyễn Thảo Mi	12/2/1987	Nữ	120, Thành Thái, Q10, TP. HCM

LOPHOC (MALH, TENLH, NGAYBD, NGAYKT, MACC, HOCPHI)

MALH	TENLH	NGAYBD	NGAYKT	MACC	HOCPHI
CCA101	Chứng chỉ A 2,4,6	25/04/2016	30/05/2016	CCA1	500000
CCA202	Chứng chỉ A 3,5,7	26/04/2016	15/06/2016	CCA2	600000
CCB101	Chứng chỉ B 2,4,6	07/07/2016	20/08/2016	CCB1	650000
CCB201	Chứng chỉ B 3,5,7	08/07/2016	10/09/2016	CCB2	700000

DANGKY (MAHV, MALH, NGAYDK, TIENDATCOC, TIENCONLAI)

MAHV	MALH	NGAYDK	TIENDATCOC	TIENCONLAI
HV001	CCA101	24/04/2016	500000	0
HV002	CCA202	26/04/2016	600000	0
HV003	CCB101	07/07/2016	500000	150000
HV004	CCB201	08/07/2016	500000	200000

Trường ĐH CNTP TP.HCM
Khoa: CNTT
Bộ môn: HTTT
MSMH: 01201009

CHƯƠNG 2
TRUY VẤN CƠ SỞ DỮ LIỆU



MỤC TIÊU

- Tạo và thực thi các lệnh cập nhật dữ liệu trên bảng.
- Tạo và thực thi câu truy vấn SQL.
- Sử dụng một số hàm thông dụng trong truy vấn.

DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SINH VIÊN

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Máy tính cài đặt SQL Server 2012	1	1	

NỘI DUNG THỰC HÀNH

PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH

Trong phần này sử dụng cơ sở dữ liệu QL_NHANVIEN để minh họa cho các câu truy vấn. Tham khảo chi tiết các lược đồ quan hệ trong cơ sở dữ liệu **QL_NHANVIEN** (*Chương 1. Phần 2. Bài tập 3*).

Lưu ý: Để thực hành theo những ví dụ trong phần này, cần phải tạo cơ sở dữ liệu QL_NHANVIEN chứa các bảng dữ liệu và nhập liệu vào bảng. Khoá chính và khoá ngoại các bảng tham khảo dưới đây:

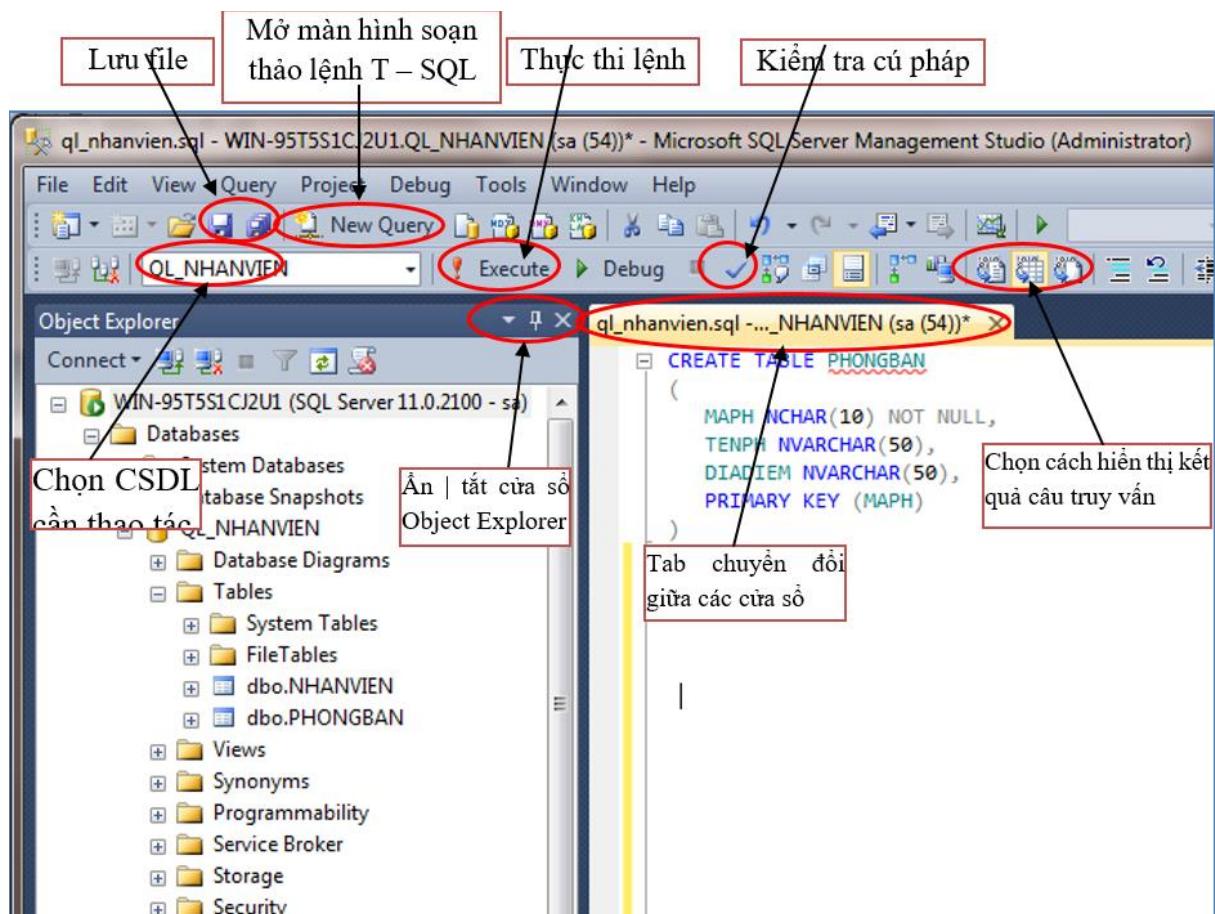
Tên bảng	Khoá chính	Khoá ngoại
PHONGBAN	MAPH	
NHANVIEN	MANV	MANQL MAPH
DEAN	MADA	
PHANCONG	MANV, MADA, THOIGIAN	MANV MADA
THANNHAN	MANV, TENTN	MANV

❖ Giới thiệu trình soạn thảo lệnh

Trình soạn thảo lệnh là môi trường để người sử dụng soạn thảo tất cả các lệnh T – SQL tương tác với cơ sở dữ liệu và hệ thống. Cũng giống như một số ngôn ngữ lập trình, khi một lệnh đã hoàn tất thì người sử dụng phải thực hiện kiểm tra lỗi cú pháp và thực thi (hoặc thực thi không cần kiểm tra lỗi cú pháp) thì lệnh đó mới được thực thi.

❖ Các thao tác cơ bản

Một số thao tác cơ bản được thể hiện trong hình sau:



Hình 2.1. Màn hình giới thiệu trình soạn thảo lệnh SQL

2.1. Cập nhật dữ liệu

Thao tác trên dữ liệu là một việc làm xảy ra thường xuyên đối với cơ sở dữ liệu. Các thao tác cập nhật sau làm ảnh hưởng trực tiếp đến dữ liệu: thêm (*INSERT*), xoá (*DELETE*), sửa (*UPDATE*).

2.1.1. Thêm dữ liệu

Xem trong nội dung nhập dữ liệu vào bảng (mục 1.8).

2.1.2. Xóa dữ liệu

Có hai cách xoá dữ liệu:

❖ Sử dụng công cụ trên SQL Server Management Studio:

Mở bảng cần xoá → chọn các dòng cần xoá → nhấn chuột phải vào vùng chọn → chọn *Delete* → chọn *Yes*.

	MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
	NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL
	NV0002	Nguyễn Minh Tâm	1990-02-10	Nam	TP. HCM	33000000.0000	NV0009	PH003
	NV0003	Nguyễn Thị Châu	1979-10-12	Nữ	VUNG TAU	50000000.0000	NV0006	PH003
	NV0004	Trần Văn Út	1977-08-23	Nam	TP. HCM	8000000.0000	NV0005	PH002
	NV0005	Trương Lệ Quyên	1987-11-22	Nữ	Hà Nội	3500000.0000	NV0008	PH002
	NV0006	Bùi Đức Chí	1987-11-22	Nam	Hà Nội	10000000.0000	NV0002	PH003
	NV0007	Nguyễn Tuấn Anh	1991-09-06	Nam	TP. HCM	10500000.0000	NV0002	PH003
	NV0008	Đỗ Xuân Thủy	1985-05-14	Nữ	Tây Ninh	150000000.0000	NV0009	PH002
	NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	18000000.0000	NULL	PH001
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

❖ Sử dụng lệnh T – SQL:

Cú pháp:

```
DELETE FROM <Tên_bảng>
[WHERE <Điều kiện xoá>]
```

Ví dụ 2.1: Xoá những nhân viên thuộc phòng có mã là PH002.

```
DELETE FROM NHANVIEN
WHERE MAPH = 'PH002';
```

Kết quả sau khi xoá:

	MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
1	NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL
2	NV0002	Nguyễn Minh Tâm	1990-02-10	Nam	TP. HCM	30000000.00	NV0009	PH003
3	NV0003	Nguyễn Thị Châu	1979-10-12	Nữ	VUNG TAU	50000000.00	NV0006	PH003
4	NV0006	Bùi Đức Chí	1987-11-22	Nam	Hà Nội	10000000.00	NV0002	PH003
5	NV0007	Nguyễn Tuấn Anh	1991-09-06	Nam	TP. HCM	10500000.00	NV0002	PH003
6	NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	18000000.00	NULL	PH001

2.1.3. Sửa dữ liệu

Có hai cách cập nhật dữ liệu:

❖ Sử dụng công cụ trên SQL Server Management Studio:

Mở bảng cần chỉnh sửa > sửa giá trị cũ thành giá trị mới.

❖ Sử dụng lệnh T – SQL:

Cú pháp:

```
UPDATE <Tên_bảng>
SET <Tên_cột> = <Giá trị | Biểu thức | Câu truy vấn>
[WHERE <Điều kiện>]
```

Ví dụ 2.2: Thay đổi địa chỉ của nhân viên có mã NV0007 từ TP. HCM thành Hà Nội.

```
UPDATE NHANVIEN
SET DIACHI = N'Hà Nội'
WHERE MANV = 'NV0007';
```

Kết quả sau khi sửa:

MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL
NV0002	Nguyễn Minh Tâm	1990-02-10	Nam	TP. HCM	33000000.0000	NV0009	PH003
NV0003	Nguyễn Thị Châu	1979-10-12	Nữ	VUNG TAU	50000000.0000	NV0006	PH003
NV0006	Bùi Đức Chí	1987-11-22	Nam	Hà Nội	10000000.0000	NV0002	PH003
NV0007	Nguyễn Tuấn Anh	1991-09-06	Nam	TP. HCM	10500000.0000	NV0002	PH003
NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	18000000.0000	NULL	PH001

2.2. Truy vấn dữ liệu

Truy vấn dữ liệu là đưa ra một yêu cầu liên quan đến dữ liệu được lưu trong CSDL. Các yêu cầu thường được đưa ra truy vấn dữ liệu bao gồm:

- ✓ Sắp xếp bản ghi.
- ✓ Chọn các trường để hiển thị.
- ✓ Tổng hợp và hiển thị dữ liệu từ nhiều bảng.
- ✓ Chọn các bản ghi thỏa mãn từ điều kiện cho trước.
- ✓ Thực hiện tính toán: Trung bình cộng, tổng, đếm,...

Cú pháp tổng quát của câu truy vấn SQL:

```

SELECT      <Các cột theo yêu cầu>
FROM        <Tên_bảng| View| Câu truy vấn>
[WHERE]     <Điều kiện trên dòng>]
[GROUP BY]   <Các cột cần nhóm>]
[HAVING]    <Điều kiện trên nhóm>]
[ORDER BY]   <Các cột cần sắp xếp>]
```

2.2.1. Cú pháp câu truy vấn

Cấu trúc câu truy vấn đơn giản:

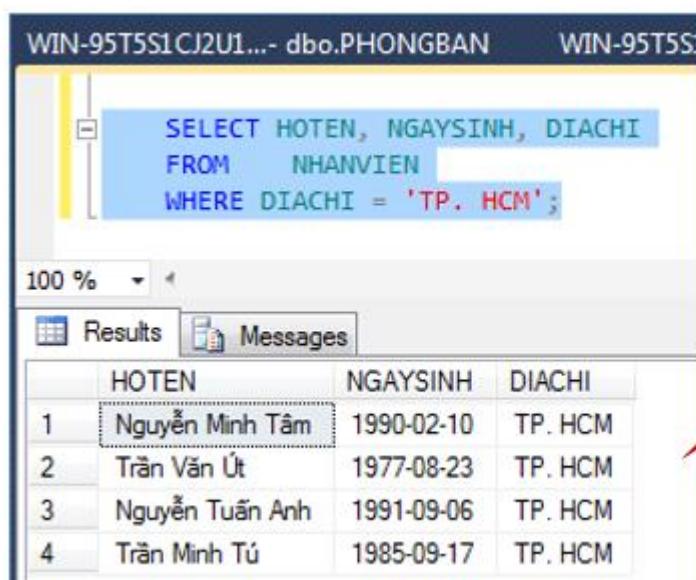
```

SELECT      <Các cột theo yêu cầu>
FROM        <Tên_bảng| View| Câu truy vấn>
[WHERE]     <Điều kiện trên dòng>]
```

Các bước thực hiện truy vấn:

1. Mở cửa sổ lệnh: Nhấn chọn biểu tượng  New Query
2. Phân tích câu truy vấn để xác định yêu cầu:
 - ✓ Mệnh đề SELECT: Chọn các thông tin cần hiển thị.
 - ✓ Mệnh đề FROM: Các bảng có liên quan.
 - ✓ Mệnh đề WHERE: Điều kiện truy vấn.
3. Sau khi xác định xong câu truy vấn thì gõ vào cửa sổ lệnh. Chọn khôi lệnh truy vấn và thực thi. Nếu không có lỗi xảy ra thì hiển thị kết quả tương ứng, ngược lại thì thông báo lỗi.

Ví dụ 2.3: Cho biết họ tên, ngày sinh, địa chỉ của những nhân viên ở TP. HCM.



The screenshot shows the SQL Server Management Studio interface. The title bar says "WIN-95T5S1CJ2U1... - dbo.PHONGBAN". The main area contains a query window with the following SQL code:

```
SELECT HOTEN, NGAYSINH, DIACHI  
FROM NHANVIEN  
WHERE DIACHI = 'TP. HCM';
```

Below the query window is a results grid with the following data:

	HOTEN	NGAYSINH	DIACHI
1	Nguyễn Minh Tâm	1990-02-10	TP. HCM
2	Trần Văn Út	1977-08-23	TP. HCM
3	Nguyễn Tuấn Anh	1991-09-06	TP. HCM
4	Trần Minh Tú	1985-09-17	TP. HCM

A red speech bubble points to the results grid with the text "Kết quả truy vấn".

2.2.2. Sử dụng tên bí danh, *

Sử dụng tên bí danh:

Tên bí danh được dùng để thay thế tên cột trong kết quả trả về khi thực thi câu truy vấn.

Cú pháp:

```
Tên_cột|Tên_bảng AS <Tên bí danh>
```

Hoặc

```
Tên_cột|Tên_bảng <Tên bí danh>
```

Ví dụ 2.4: Cho biết họ tên, ngày sinh, mã người quản lý được đặt bí danh là “Người quản lý” của những nhân viên nữ.

```

WIN-95T5S1CJ2U1...- dbo.PHONGBAN      WIN-95T5S1CJ2U1...N - dbo.N
[SELECT HOTEN, NGAYSINH, MANQL AS N'Người quản lý'
  FROM NHANVIEN
 WHERE PHAI = N'Nữ'];
100 % < >
Results Messages
HOTEN NGAYSINH Người quản lý
1 Nguyễn Thị Châu 1979-10-12 NV006
2 Trương Lê Quyên 1987-11-22 NV008
3 Đỗ Xuân Thùy 1985-05-14 NV009

```

Tên bí danh

Sử dụng ký tự *

Khi muốn câu truy vấn trả về tất cả các cột trong bảng thì dùng ký tự *.

Ví dụ 2.5: Cho biết thông tin tất cả các phòng ban.

Hai câu truy vấn sau là tương đương:

```

ql_nhanvien.sql -..._NHANVIEN (sa (53))*
SELECT * FROM PHONGBAN;

[SELECT MAPH, TENPH, DIADIEM
  FROM PHONGBAN];
100 % < >
Results Messages
MAPH TENPH DIADIEM
1 PH001 Kế hoạch Tầng 1 nhà A
2 PH002 Quản trị Tầng 1 nhà B
3 PH003 Nhân sự Tầng 2 nhà A
4 PH004 Tài vụ Tầng 3 Nhà A
5 PH005 Đầu tư Tầng 2 Nhà B

```

	MAPH	TENPH	DIADIEM
1	PH001	Kế hoạch	Tầng 1 nhà A
2	PH002	Quản trị	Tầng 1 nhà B
3	PH003	Nhân sự	Tầng 2 nhà A
4	PH004	Tài vụ	Tầng 3 Nhà A
5	PH005	Đầu tư	Tầng 2 Nhà B

	MAPH	TENPH	DIADIEM
1	PH001	Kế hoạch	Tầng 1 nhà A
2	PH002	Quản trị	Tầng 1 nhà B
3	PH003	Nhân sự	Tầng 2 nhà A
4	PH004	Tài vụ	Tầng 3 Nhà A
5	PH005	Đầu tư	Tầng 2 Nhà B

Trong trường hợp câu truy vấn có liên quan đến nhiều bảng nhưng ta chỉ muốn lấy tất cả các cột trên một bảng thì sử dụng cú pháp: <Tên_bảng>. *

Ví dụ 2.6: Cho biết thông tin những nhân viên thuộc phòng Kế hoạch. Yêu cầu kết quả chỉ lấy tất cả các cột trên bảng NHANVIEN.

```
SELECT NHANVIEN.*  
FROM NHANVIEN, PHONGBAN  
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH AND TENPH = N'Kế hoạch';
```

	MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
1	NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	18000000.00	NULL	PH001

2.2.3. Xử lý chuỗi bằng toán tử Like

Trường hợp một câu truy vấn liên quan đến việc xử lý chuỗi thì dùng toán tử LIKE.

Cú pháp:

```
<Tên_cột> LIKE '%chuỗi' | '_chuỗi'  
<Tên_cột> LIKE 'chuỗi%' | 'chuỗi_'  
<Tên_cột> LIKE '%chuỗi%' | '_chuỗi_'
```

Trong đó:

- ✓ Ký tự % đại diện cho nhiều ký tự bất kỳ.
- ✓ Ký tự _ đại diện cho một ký tự bất kỳ.

Ví dụ 2.7: Cho biết họ tên những nhân viên có họ “Nguyễn”.

Nhận xét: “Nguyễn” là chuỗi nằm ở vị trí đầu trong chuỗi họ tên, do đó những nhân viên nào có họ “Nguyễn” là hợp lệ.

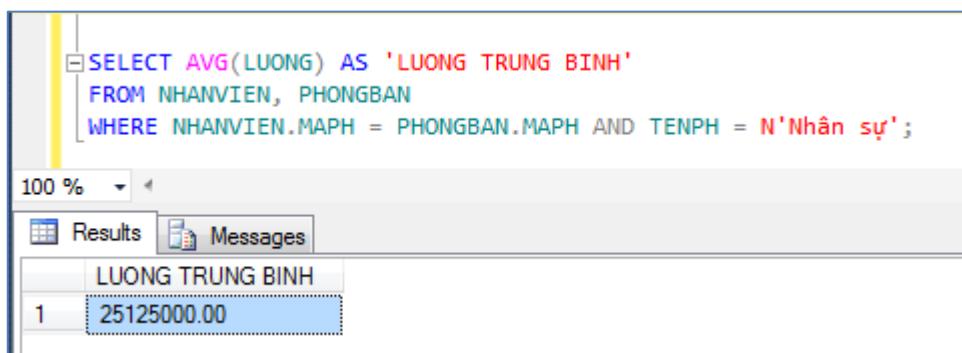
```
SELECT HOTEN  
FROM NHANVIEN  
WHERE HOTEN LIKE N'Nguyễn%';
```

HOTEN
1 Nguyễn Văn Nam
2 Nguyễn Minh Tâm
3 Nguyễn Thị Châu
4 Nguyễn Tuấn Anh

2.2.4. Các hàm thống kê

Một số hàm thống kê: Count, Sum, AVG, Max, Min.

Ví dụ 2.8: Cho biết mức lương trung bình của các nhân viên thuộc phòng Nhân sự.



```
SELECT AVG(LUONG) AS 'LUONG TRUNG BINH'
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH AND TENPH = N'Nhân sự';

100 % ▾
Results Messages
LUONG TRUNG BINH
1 25125000.00
```

Lưu ý:

- ✓ Các hàm thống kê chỉ được sử dụng trong mệnh đề **Select** hay mệnh đề **Having**, không được sử dụng trong mệnh đề **Where**.
- ✓ Các hàm thống kê không được viết lồng nhau.

❖ Mệnh đề gom nhóm

Khi cần nhóm các dòng có dữ liệu giống nhau tại một hay nhiều thuộc tính để thực hiện tính toán thì sử dụng mệnh đề gom nhóm Group By.

Cú pháp:

SELECT	<Các cột theo yêu cầu>
FROM	<Tên bảng View Câu truy vấn>
[WHERE]	<Điều kiện trên dòng>】
GROUP BY	<Các cột cần nhóm>
[HAVING]	<Điều kiện trên nhóm>】

Ví dụ 2.9: Liệt kê tên phòng và số lượng nhân viên tương ứng.

The diagram illustrates a many-to-many relationship between the **NHANVIEN** and **PHONGBAN** tables. An arrow points from the **MAPH** column in **NHANVIEN** to the **MAPH** column in **PHONGBAN**, indicating that an employee can be assigned to multiple departments. Conversely, an arrow points from the **MAPH** column in **PHONGBAN** back to the **MAPH** column in **NHANVIEN**, indicating that a department can have multiple employees assigned to it. The query results show the following data:

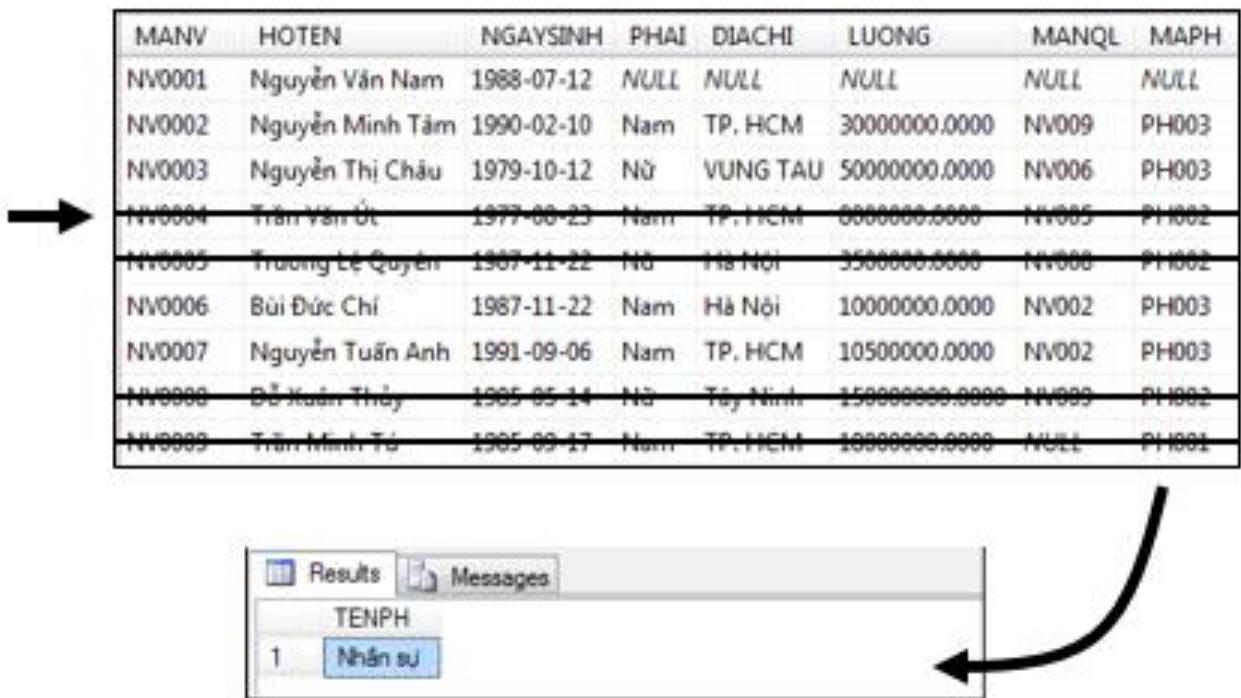
TENPH	SO LUONG NHAN VIEN
Kế hoạch	1
Nhân sự	4
Quản trị	3

MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL
NV0002	Nguyễn Minh Tâm	1990-02-10	Nam	TP. HCM	30000000.0000	NV009	PH003
NV0003	Nguyễn Thị Châu	1979-10-12	Nữ	VUNG TAU	50000000.0000	NV006	PH003
NV0004	Trần Văn Út	1977-08-23	Nam	TP. HCM	8000000.0000	NV005	PH002
NV0005	Trương Lệ Quyên	1987-11-22	Nữ	Hà Nội	3500000.0000	NV008	PH002
NV0006	Bùi Đức Chí	1987-11-22	Nam	Hà Nội	10000000.0000	NV002	PH003
NV0007	Nguyễn Tuấn Anh	1991-09-06	Nam	TP. HCM	10500000.0000	NV002	PH003
NV0008	Đỗ Xuân Thủy	1985-05-14	Nữ	Tây Ninh	150000000.0000	NV009	PH002
NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	18000000.0000	NULL	PH001

Sau khi gom nhóm bằng lệnh Group By, nếu có điều kiện trên nhóm thì điều kiện này được đặt trong mệnh đề Having.

Ví dụ 2.10: Liệt kê tên những phòng có số lượng nhân viên lớn hơn 3.

```
SELECT TENPH
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH
GROUP BY TENPH
HAVING COUNT(*) > 3;
```



The diagram illustrates a query execution process. At the top, a table shows nine rows of employee data. An arrow points from this table to a SQL results window below it. The results window has tabs for 'Results' and 'Messages'. The 'Results' tab displays a single row: 'TENPH' followed by the value 'Nhân sự'.

MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
NV0001	Nguyễn Văn Nam	1988-07-12	NULL	NULL	NULL	NULL	NULL
NV0002	Nguyễn Minh Tâm	1990-02-10	Nam	TP. HCM	30000000.0000	NV009	PH003
NV0003	Nguyễn Thị Châu	1979-10-12	Nữ	VUNG TAU	50000000.0000	NV006	PH003
NV0004	Trần Văn Út	1977-08-23	Nam	TP. HCM	8000000.0000	NV005	PH002
NV0005	Trương Lê Quyết	1987-11-22	Nő	Hà Nội	3500000.0000	NV000	PH002
NV0006	Bùi Đức Chí	1987-11-22	Nam	Hà Nội	10000000.0000	NV002	PH003
NV0007	Nguyễn Tuấn Anh	1991-09-06	Nam	TP. HCM	10500000.0000	NV002	PH003
NV0008	Đỗ Xuân Thủy	1985-05-14	Nő	Tây Ninh	15000000.0000	NV003	PH002
NV0009	Trần Minh Tú	1985-09-17	Nam	TP. HCM	10000000.0000	NULL	PH001

Lưu ý:

- ✓ Các thuộc tính trong mệnh đề Group By phải có mặt trong mệnh đề Select.
- ✓ Các hàm thống kê không có mặt trong mệnh đề Group By
- ❖ **Sắp xếp các bộ**

Khi thực thi câu truy vấn, đôi khi kết quả trả về cần được sắp xếp theo một trật tự nào đó. Ví dụ: sắp xếp họ tên sinh viên theo thứ tự tăng dần, sắp xếp điểm trung bình sinh viên theo thứ tự giảm dần,... Các cột cần sắp xếp được đặt trong mệnh đề ORDER BY, có hai cách sắp xếp là tăng dần và giảm dần.

Cú pháp:

```

SELECT      <Các cột theo yêu cầu>
FROM        <Tên bảng| View| Câu truy vấn>
[WHERE]     <Điều kiện trên dòng>
ORDER BY    <Các cột cần sắp xếp> [ASC| DESC]
  
```

Ví dụ 2.11: Cho biết họ tên, năm sinh, địa chỉ của những nhân viên thuộc phòng Nhân sự hoặc Quản trị, sắp xếp năm sinh theo thứ tự tăng dần.

```

SELECT HOTEN, YEAR(NGAYSINH) 'NAM SINH', DIACHI
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH AND (TENPH = N'Nhân sự' OR TENPH = N'Quản trị')
ORDER BY YEAR(NGAYSINH) ASC;

```

Results

	HOTEN	NAM SINH	DIACHI
1	Trần Văn Út	1977	TP. HCM
2	Nguyễn Thị Châu	1979	VŨNG TÀU
3	Đỗ Xuân Thùy	1985	Tây Ninh
4	Trương Lệ Quyên	1987	Hà Nội
5	Bùi Đức Chí	1987	Hà Nội
6	Nguyễn Minh Tâm	1990	TP. HCM
7	Nguyễn Tuấn Anh	1991	TP. HCM

Ngoài ra, có thể sắp xếp cùng lúc nhiều cột. Trong trường hợp này, cột ghi trước có độ ưu tiên cao hơn và được sắp xếp trước, nếu có các giá trị trùng nhau thì sắp xếp dựa vào cột có độ ưu tiên thấp hơn, cứ như vậy cho đến khi sắp xếp xong.

Xét ví dụ như sau:

```

SELECT HOTEN, YEAR(NGAYSINH) 'NAM SINH', TENPH
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH AND (TENPH = N'Nhân sự' OR TENPH = N'Quản trị')
ORDER BY YEAR(NGAYSINH) ASC, TENPH DESC;

```

Results

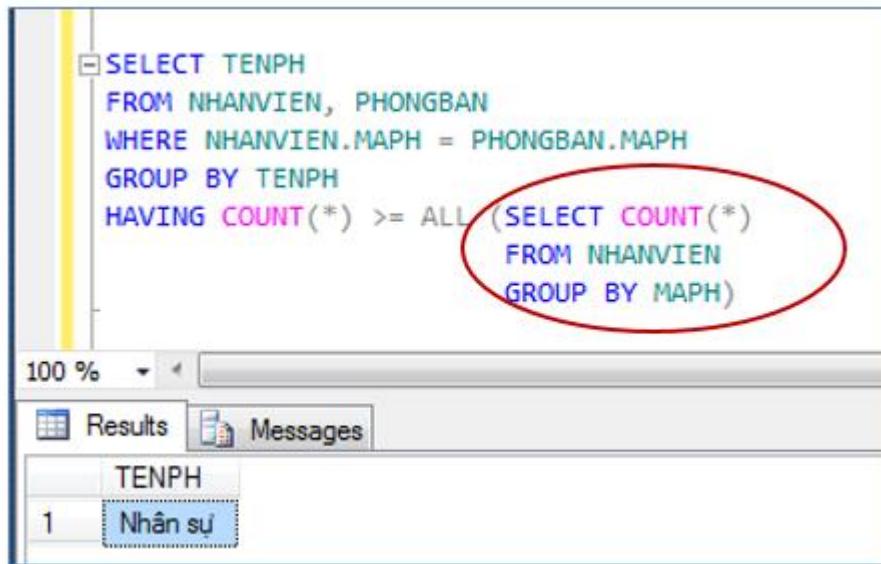
	HOTEN	NAM SINH	TENPH
1	Trần Văn Út	1977	Quản trị
2	Nguyễn Thị Châu	1979	Nhân sự
3	Đỗ Xuân Thùy	1985	Quản trị
4	Trương Lệ Quyên	1987	Quản trị
5	Bùi Đức Chí	1987	Nhân sự
6	Nguyễn Minh Tâm	1990	Nhân sự
7	Nguyễn Tuấn Anh	1991	Nhân sự

2.3. Truy vấn nâng cao

2.3.1. Truy vấn lồng

Truy vấn lồng (truy vấn con) là lồng câu truy vấn này trong câu truy vấn khác để xử lý các yêu cầu truy vấn phức tạp. Các từ khoá thường dùng trong truy vấn con: IN|NOT IN, ALL, SOME|ANY, EXISTS|NOT EXISTS. Khi thực thi câu lệnh, câu truy vấn con được xử lý trước, kết quả được sử dụng cho câu truy vấn cha xử lý tiếp và xuất ra kết quả cuối cùng.

Ví dụ 2.12: Cho biết tên phòng ban có nhiều nhân viên nhất.



```
SELECT TENPH
FROM NHANVIEN, PHONGBAN
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH
GROUP BY TENPH
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
                           FROM NHANVIEN
                           GROUP BY MAPH)
```

The screenshot shows the SQL query above and its results. The results table has one row with column headers 'TENPH' and data 'Nhân sự'. A red oval highlights the subquery in the HAVING clause.

TENPH
Nhân sự

2.3.2. Truy vấn với các phép toán tập hợp (UNION, INTERSECT, EXCEPT)

❖ Kết hợp các kết quả với toán tử UNION

UNION cho phép kết hợp hai hoặc nhiều bộ kết quả từ nhiều bảng với nhau.

Cú pháp:

```
SELECT <danh sách cột>
UNION [DISTINCT | ALL]
SELECT <danh sách cột>
UNION [DISTINCT | ALL]
...
```

Lưu ý: Khi sử dụng UNION

- ✓ Số lượng các cột trong mỗi câu lệnh SELECT phải giống nhau.
- ✓ Các kiểu dữ liệu của cột trong danh sách cột của câu lệnh SELECT phải giống nhau hoặc ít nhất là có thể chuyển đổi sang cho nhau.

Ví dụ 2.13: Kết hợp thông tin về nhân viên và thân nhân thành một tập hợp kết quả.

The screenshot shows a SQL query window titled "ql_nhanvien.sql -..._NHANVIEN (sa (56))". The query is:

```
SELECT MANV 'MA NV', HOTEN 'HO TEN'  
FROM NHANVIEN  
UNION  
SELECT MANV 'MA NV', TENTN 'HO TEN'  
FROM THANHNGHAN;
```

The results pane shows a table with two columns: "MA NV" and "HO TEN". The data is as follows:

	MA NV	HO TEN
1	NV0001	Nguyễn Thị Tâm
2	NV0001	Nguyễn Văn Bình
3	NV0001	Nguyễn Văn Nam
4	NV0002	Nguyễn Chính Nghĩa
5	NV0002	Nguyễn Minh Tâm
6	NV0003	Nguyễn Thị Châu
7	NV0004	Trần Văn Út
8	NV0005	Lê Anh Hùng
9	NV0005	Trường Lê Quyên
10	NV0006	Bùi Đại An
11	NV0006	Bùi Đức Chí
12	NV0007	Nguyễn Tuấn Anh

Khi sử dụng ORDER BY để sắp xếp kết quả với UNION, phải đặt nó ở vị trí cuối cùng trong mệnh đề SELECT.

Ví dụ 2.14: Kết hợp thông tin về nhân viên và thân nhân thành một tập hợp kết quả. Sau đó sắp xếp kết quả theo họ tên và mã thứ tự tăng dần.

The screenshot shows a SQL query window titled "ql_nhanvien.sql -..._NHANVIEN (sa (56))". The query is:

```
SELECT MANV, HOTEN  
FROM NHANVIEN  
UNION  
SELECT MANV, TENTN  
FROM THANHNGHAN  
ORDER BY HOTEN, MANV;
```

The results pane shows a table with two columns: "MANV" and "HOTEN". The data is as follows:

	MANV	HOTEN
1	NV0006	Bùi Đại An
2	NV0006	Bùi Đức Chí
3	NV0008	Đỗ Xuân Thúy
4	NV0005	Lê Anh Hùng
5	NV0008	Lê Thảo Nguyên
6	NV0002	Nguyễn Chính Nghĩa
7	NV0002	Nguyễn Minh Tâm
8	NV0003	Nguyễn Thị Châu
9	NV0001	Nguyễn Thị Tâm
10	NV0007	Nguyễn Tuấn Anh
11	NV0001	Nguyễn Văn Bình
12	NV0001	Nguyễn Văn Nam

Nếu tên cột không giống nhau trong hai mệnh đề SELECT của phép UNION và không sử dụng bí danh cho mỗi cột trong mệnh đề SELECT thì SQL Server sẽ sử dụng tên cột của câu lệnh SELECT đầu tiên là tên cột trong kết quả đầu ra.

Có thể sắp xếp các kết quả thiết lập dựa trên vị trí cột trong mệnh đề ORDER BY. Kết quả của phép UNION giữa hai tập kết quả từ bảng dữ liệu NHANVIEN và THANNHAN trong ví dụ 2.14 có thể viết như sau:

```
SELECT MANV, HOTEN  
FROM NHANVIEN  
UNION  
SELECT MANV, TENTN  
FROM THANNHAN  
ORDER BY 2, 1;
```

❖ Truy vấn INTERSECT

Intersect kết hợp tập kết quả của hai hoặc nhiều truy vấn và trả về các bản ghi xuất hiện trong cả hai truy vấn.

Cú pháp:

```
QUERY 1  
INTERSECT  
QUERY 2
```

Lưu ý: Khi sử dụng INTERSECT

- ✓ Cả hai truy vấn phải có cùng số lượng và thứ tự cột.
- ✓ Kiểu dữ liệu của các cột tương ứng phải giống nhau

Ví dụ 2.15: Cho biết danh sách nhân viên (MANV) được phân công?

The screenshot shows a SQL Server Management Studio window. In the top pane, there is a tree view with a node expanded to show two queries: "Select MANV from NHANVIEN" and "Select MANV from PHANCONG". Below this, the text "INTERSECT" is typed. In the bottom pane, there are two tabs: "Results" and "Messages". The "Results" tab is selected and displays a table with three rows. The table has one column labeled "MANV". The data is as follows:

MANV
1 NV0003
2 NV0005
3 NV0007

❖ Truy vấn EXCEPT

Except so sánh tập hợp kết quả của hai truy vấn và trả về các bản ghi của truy vấn thứ nhất không xuất hiện trong các bản ghi của truy vấn thứ hai

Cú pháp:

```
QUERY 1  
EXCEPT  
QUERY 2
```

Ví dụ 2.16: Cho biết danh sách nhân viên (MANV) chưa được phân công?

```
Select MANV from NHANVIEN  
EXCEPT  
Select MANV from PHANCONG
```

MANV
1 NV00013
2 NV0002
3 NV0004
4 NV0006
5 NV0008
6 NV0009

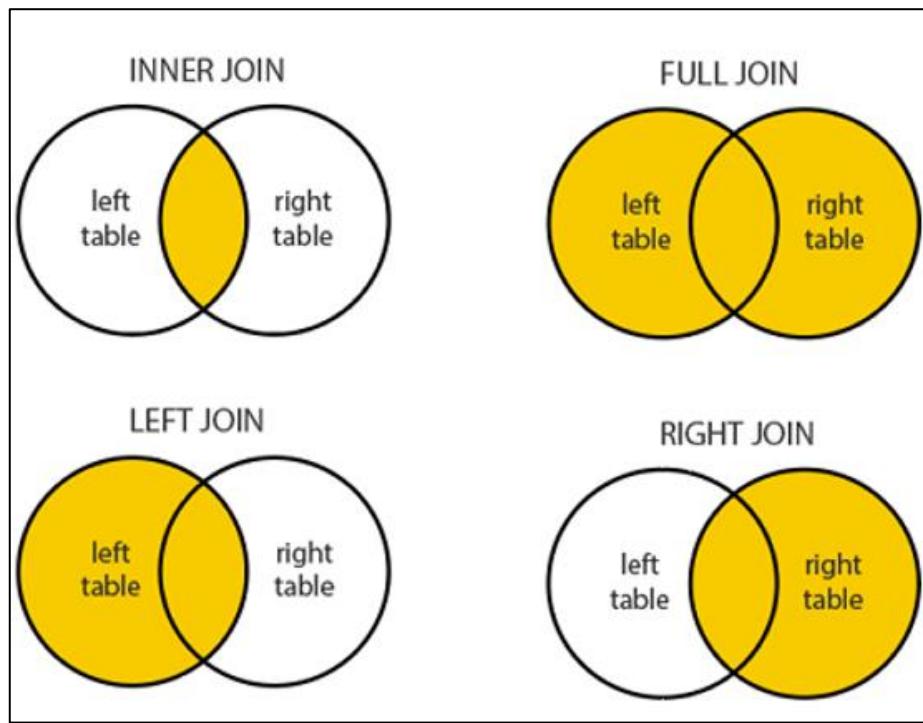
Lưu ý: Khi sử dụng EXCEPT

- ✓ Số lượng và thứ tự của các cột phải giống nhau trong cả hai truy vấn
- ✓ Các kiểu dữ liệu của các cột tương ứng phải giống nhau hoặc tương thích

2.3.3. Truy vấn với phép kết ngoài (outer join)

Phép kết nối dùng để liên kết các bảng có quan hệ với nhau trong cơ sở dữ liệu. Giữa hai bảng kết nối phải có thuộc tính chung với nhau. Thứ tự kết nối của các bảng có ảnh hưởng đến kết quả truy vấn.

Trong tài liệu này đề cập đến các phép nối sau: phép nối trong (Inner Join), phép nối trái (Left Join), phép nối phải (Right Join),



Hình 2. 2. Mô tả phép kết nối Join

2.3.3.1. Inner Join

Phép nối trong là một phần tùy chọn của câu lệnh SELECT, xuất hiện ngay sau mệnh đề FROM. Bảng cần kết nối với bảng chính phải xuất hiện sau từ khoá Inner Join. Điều kiện kết nối xuất hiện sau từ khoá ON.

Cú pháp:

```

SELECT <Danh sách cột>
FROM <Tên bảng>
INNER JOIN <Tên bảng quan hệ>
ON <Điều kiện kết nối>;
    
```

Giả sử nối hai bảng A và B, INNER JOIN so sánh mỗi bản ghi của bảng A với mỗi bản ghi của bảng B để tìm tất cả các cặp bản ghi đáp ứng điều kiện nối. Khi điều kiện nối được thỏa mãn, giá trị cột cho mỗi cặp bản ghi phù hợp của bảng A và bảng B được kết hợp thành một bản ghi trong kết quả trả về.

Hạn chế sự trùng tên cột khi sử dụng INNER JOIN: Nếu nối nhiều bảng có cột cùng tên, phải chỉ rõ tên bảng có chứa cột dữ liệu định lấy để tránh lỗi cột không rõ ràng. Giả sử bảng A và B đều có cột M. Trong câu lệnh SELECT với INNER JOIN, phải tham chiếu tới cột M bằng cách sử dụng cú pháp A.M.

Ví dụ 2.17: Cho biết thông tin nhân viên ở phòng Quản trị.

```
SELECT NV.*  
FROM NHANVIEN NV  
INNER JOIN PHONGBAN PB ON NV.MAPH = PB.MAPH  
WHERE TENPH = N'Quản trị';
```

	MANV	HOTEN	NGAYSINH	PHAI	DIACHI	LUONG	MANQL	MAPH
1	NV0004	Trần Văn Út	1977-08-23	Nam	TP. HCM	8000000.00	NV005	PH002
2	NV0005	Trương Lê Quyên	1987-11-22	Nữ	Hà Nội	3500000.00	NV008	PH002
3	NV0008	Đỗ Xuân Thùy	1985-05-14	Nữ	Tây Ninh	150000000.00	NV009	PH002

INNER JOIN so sánh từng dòng trong bảng NHANVIEN và PHONGBAN để tìm một cặp bản ghi có cùng mã phòng (MAPH). Nếu một cặp bản ghi có cùng mã phòng khi đó thông tin nhân viên cũng sẽ được kết hợp thành một hàng để trả lại kết quả.

2.3.3.2. Left outer Join

Phép nối trái cũng là một phần tùy chọn của câu lệnh SELECT, cho phép lấy thêm dữ liệu từ các bảng khác.

Khi một dòng dữ liệu trong bảng bên trái mệnh đề Left Join phù hợp với một dòng trong bảng bên phải dựa trên điều kiện kết nối, nội dung của dòng đó sẽ kết hợp với dòng trong bảng bên phải hiển thị trong kết quả đầu ra. Nếu một dòng trong bảng bên trái không tìm được dòng nào phù hợp trong bảng bên phải, nó vẫn xuất hiện trong kết quả đầu ra nhưng kết hợp với một dòng “giả” từ bảng bên phải với giá trị NULL cho tất cả các cột.

Như vậy, Left Join cho phép chọn tất cả các dòng dữ liệu từ bảng bên trái ngay cả khi không có bản ghi nào phù hợp với nó trong bảng bên phải.

Cú pháp:

```
SELECT <Danh sách cột>  
FROM <Tên bảng>  
LEFT [OUTER] JOIN < Tên bảng quan hệ >  
ON <Điều kiện kết nối>;
```

Ví dụ 2.18: Cho biết Họ tên của tất cả nhân viên và tình trạng phân công thực hiện đề án (MADA, THOIGIAN).

```
SELECT NV.HOTEN, PC.MADA, PC.THOIGIAN
FROM NHANVIEN NV
LEFT JOIN PHANCONG PC ON NV.MANV = PC.MANV;
```

	HOTEN	MADA	THOIGIAN
1	Nguyễn Văn Nam	DA001	2015-02-05
2	Nguyễn Văn Nam	DA003	2016-03-17
3	Nguyễn Minh Tâm	NULL	NULL
4	Nguyễn Thị Châu	DA003	2016-01-01
5	Trần Văn Út	NULL	NULL
6	Trương Lệ Quyên	DA004	2016-05-10
7	Bùi Đức Chí	NULL	NULL
8	Nguyễn Tuấn Anh	DA005	2016-12-20
9	Đỗ Xuân Thùy	NULL	NULL
10	Trần Minh Tú	NULL	NULL

Ở bảng kết quả trên, có thể thấy tất cả các nhân viên được liệt kê. Trong đó, có những bản ghi có thông tin nhân viên nhưng tất cả các thông tin về việc phân công thực hiện đề án đều có giá trị NULL. Như vậy, những nhân viên đó không được phân công vào bất kỳ đề án nào.

LEFT JOIN rất hữu ích khi muốn tìm các bản ghi trong bảng bên trái mà không phù hợp với bất kỳ một bản ghi nào trong bảng bên phải. Có thể thực hiện điều này bằng cách thêm một mệnh đề WHERE để lựa chọn các dòng chỉ có giá trị NULL trong một cột ở bảng bên phải.

Ví dụ 2.19: Tìm tất cả những nhân viên không được phân công bất kỳ đề án nào. Thông tin hiển thị gồm: họ tên nhân viên, mã đề án.

```
SELECT NV.hoten, PC.MADA
FROM NHANVIEN NV
LEFT JOIN PHANCONG PC ON NV.MANV = PC.MANV
WHERE MADA IS NULL;
```

	HOTEN	MADA
1	Nguyễn Minh Tâm	NULL
2	Trần Văn Út	NULL
3	Bùi Đức Chí	NULL
4	Đỗ Xuân Thùy	NULL
5	Trần Minh Tú	NULL

2.3.3.3. Right outer Join

Phép nối phải tương tự như phép nối trái nhưng kết hợp dữ liệu trong bảng bên phải mệnh đề Right Join với dữ liệu trong bảng bên trái mệnh đề.

Cú pháp:

```
SELECT <Danh sách cột>
FROM <Tên bảng>
RIGHT [OUTER] JOIN <Tên bảng quan hệ>
ON <Điều kiện kết nối>;
```

Ví dụ 2.20: Cho biết tên nhân viên, mã đề án của tất cả các đề án đã được phân công.

```
SELECT NV.HOTEN, PC.MADA
FROM NHANVIEN NV
RIGHT JOIN PHANCONG PC ON NV.MANV = PC.MANV;
```

	HOTEN	MADA
1	Nguyễn Văn Nam	DA001
2	Nguyễn Văn Nam	DA003
3	Nguyễn Thị Châu	DA003
4	Trương Lệ Quyên	DA004
5	Nguyễn Tuấn Anh	DA005

2.3.3.4. Full outer Join

Full outer Join là mệnh đề truy vấn với kết quả trả về là tập hợp tất cả dữ liệu chung và riêng giữa thông qua điều kiện kết hợp hai bảng.

```
SELECT <Danh sách cột>
FROM <Tên bảng>
FULL OUTER JOIN <Tên bảng quan hệ>
ON <Điều kiện kết nối>;
```

Ví dụ 2.21: Cho biết danh sách nhân viên (Họ tên) và danh sách phòng ban (mã phòng ban) tương ứng?

```
SELECT NV.HOTEN, PB.MAPH  
FROM NHANVIEN NV  
FULL OUTER JOIN PHONGBAN PB ON NV.MAPH = PB.MAPH
```

HOTEN	MAPH
Nguyễn Thị Châu	PH003
Trần Văn Út	PH002
Trần Lệ Quyên	PH002
Bùi Đức Chí	PH003
Nguyễn Tuấn Anh	PH003
Đỗ Xuân Thùy	PH002
Trần Minh Tú	PH001
Trần Khánh An	NULL
Nguyễn Ngọc Phan	NULL
NULL	PH004
NULL	PH005
NULL	PH006
NULL	PH007
NULL	PH008

2.3.3.5. Một số trường hợp khác của câu lệnh truy vấn

❖ Nhập dữ liệu sử dụng thông tin từ bảng khác

Dữ liệu cần nhập vào bảng có thể được lấy từ một bảng khác bằng cách thực thi câu lệnh SELECT. Giả sử cần lấy mã nhân viên và họ tên của những nhân viên phòng Nhân sự, khi đó ta tạo một bảng dữ liệu có tên NV_NHANSU để lưu trữ thông tin. Nhập dữ liệu vào bảng NV_NHANSU bằng cách trích ra những nhân viên này từ bảng NHANVIEN và chèn vào bảng NV_NHANSU. Thực hiện điều này bằng cách thực thi câu lệnh như sau:

```
INSERT INTO NV_NHANSU  
SELECT MANV, HOTEN  
FROM NHANVIEN, PHONGBAN  
WHERE NHANVIEN.MAPH = PHONGBAN.MAPH  
AND TENPH = N'Nhân sự';
```

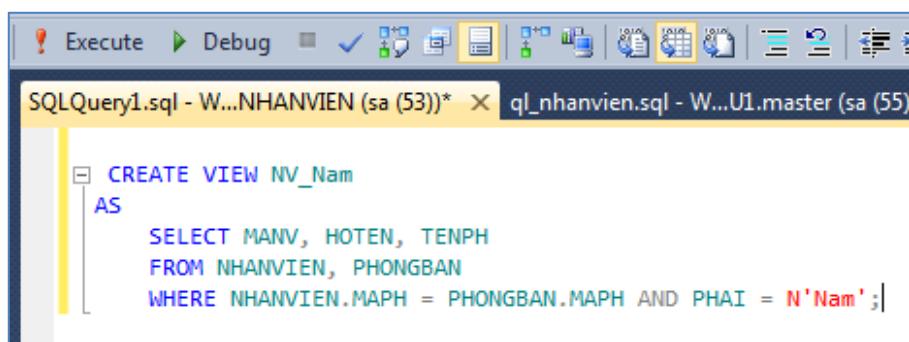
❖ **Tạo bảng ảo bằng cách sử dụng lệnh T - SQL**

Cú pháp:

```
CREATE VIEW <tên_bảng_ảo>
AS
    Câu lệnh SELECT
```

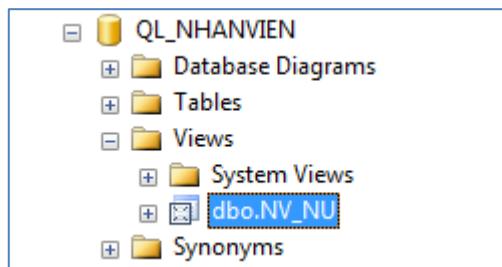
Ví dụ 2.22: Tạo bảng ảo trong ví dụ trên sử dụng T - SQL:

Mở cửa sổ soạn thảo lệnh, gõ vào đoạn lệnh tạo bảng ảo như sau:



```
CREATE VIEW NV_Nam
AS
    SELECT MANV, HOTEN, TENPH
    FROM NHANVIEN, PHONGBAN
    WHERE NHANVIEN.MAPH = PHONGBAN.MAPH AND PHAI = N'Nam';
```

Sau khi gõ xong lệnh, chọn khối lệnh vừa gõ và nhấn chọn công cụ  trên thanh công cụ để thực thi. Khi đó một bảng ảo sẽ được tạo.



PHẦN 2: BÀI TẬP THỰC HÀNH

Bài 1: Cho lược đồ cơ sở dữ liệu quản lý nhân viên gồm có các lược đồ quan hệ sau:

KHOA(MAKH, TENKH)

LOP(MALOP, TENLOP, SISO, LOPTRUONG, MAKH)

SINHVIEN(MASV, HOTEN, NGSINH, GTINH, DCHI, MALOP)

GIANGVIEN(MAGV, TENGV, MAKH)

MONHOC(MAMH, TENMH, SOTC)

DIEM(MASV, MAMH, LANTHI, DIEMTHI)

GIANGDAY(MAGV, MAMH, NAMHOC, HOCKY)

THANHMANH(MATN, HOTEN, GIOITINH)

QUANHE(MATN, MASV, QUANHE)

Yêu cầu:

- Viết lệnh tạo cơ sở dữ liệu có tên QL_SINHVIEN, tạo bảng (tạo khóa chính, khóa ngoại khi tạo bảng) và nhập liệu cho các bảng như sau:

Bảng **KHOA**

MAKH	TENKH
SH	Công nghệ sinh học
TH	Công nghệ thông tin
TP	Công nghệ thực phẩm
QT	Quản trị kinh doanh
TC	Tài chính kế toán

Bảng **LOP**

MALOP	TENLOP	SISO	LOPTRUONG	MAKH
10DHS1	10 Đại học Sinh học 1	55	SV008	SH
10DHTH1	10 Đại học Tin học 1	50	SV001	TH
11DHTH2	11 Đại học Tin học 2	40	SV005	TH
12DHTC1	12 Đại học Tài chính 1	75	SV009	TC
12DHTP1	12 Đại học Thực phẩm 1	60	SV007	TP

Bảng **SINHVIEN**

MASV	HOTEN	NGSINH	GTINH	DIACHI	MALOP
SV001	Trần Lệ Quyên	21/01/1995	Nữ	TPHCM	10DHTH1
SV002	Nguyễn Thế Bình	04/06/1996	Nam	Tây Ninh	11DHTH2
SV003	Tô Ánh Nguyệt	02/05/1995	Nữ	Vũng Tàu	12DHTP1
SV004	Nguyễn Thế Anh	15/12/1996	Nam	Đồng Nai	12DHTP1
SV005	Lê Thanh Bình	09/12/1994	Nam	Long An	10DHTH1
SV006	Phạm Quang Hậu	12/10/1995	Nam	Tây Ninh	10DHTH1
SV007	Lê Cẩm Tú	13/02/1989	Nữ	Bình Thuận	12DHTP1
SV008	Trương Thế Sang	04/04/1993	Nam	Bình Dương	10DHSH1
SV009	Đậu Quang Ánh	03/12/1994	Nam	Long An	12DHTC1
SV010	Huỳnh Kim Chi	18/10/1996	Nữ	TPHCM	11DHTH2
SV011	Trịnh Đình Ánh	15/11/1995	Nam	Bình Thuận	10DHTH1

Bảng GIANGVIEN

MAGV	TENGV	MAKH
GV001	Phạm Thế Bảo	TH
GV002	Lê Thè Truyền	TH
GV003	Trương Anh Dũng	SH
GV004	Bùi Chí Anh	TC
GV005	Lê Công Hậu	QT
GV006	Lê Trung Thành	TP

Bảng MONHOC

MAMH	TENMH	SOTC
CSDL	Cơ sở dữ liệu	3
KTLT	Kỹ thuật lập trình	3
THVP	Tin học văn phòng	3
TRR	Toán rời rạc	3
TTNT	Trí tuệ nhân tạo	2
TTQT	Thanh toán quốc tế	2

Bảng DIEM

MASV	MAMH	LANTHI	DIEMTHI
SV001	CSDL	1	9
SV002	THVP	1	3
SV002	THVP	2	7
SV004	THVP	1	6
SV004	TTQT	1	5
SV005	CSDL	1	3
SV005	CSDL	2	6
SV006	KTLT	1	4
SV009	TTQT	1	4
SV010	THVP	1	8
SV010	TRR	1	7

Bảng GIANGDAY

MASV	MAMH	LANTHI	DIEMTHI
GV001	CSDL	2021-2022	1
GV001	KTLT	2020-2021	2
GV001	TTNT	2020-2021	1
GV002	CSDL	2021-2022	2
GV002	KTLT	2021-2022	2

Bảng THANHANH

MATN	HOTEN	GIOITINH
TN001	Nguyễn Thế Thành	Nam
TN002	Tô Ánh Hồng	Nữ
TN003	Lê Thanh An	Nam
TN004	Phạm Thanh Tiễn	Nữ
TN006	Đậu Văn Thanh	Nam
TN007	Nguyễn Thị Ánh	Nữ
TN008	Lê Quang Định	Nam
TN009	Huỳnh Văn Tư	Nam

Bảng QUANHE

MATN	MASV	GIOITINH
TN001	SV002	Bố
TN003	SV005	Bố
TN004	SV007	Mẹ
TN006	SV009	Bố
TN007	SV002	Mẹ
TN008	SV005	Bố
TN008	SV007	Bố

2. Cập nhật địa chỉ của sinh viên có mã số SV002 là ‘CẦN THƠ’.
3. Tạo bảng ảo lưu trữ danh sách sinh viên (Mã SV, họ tên), tên lớp tương ứng của sinh viên.
4. Liệt kê danh sách sinh viên (Mã SV, Họ tên) của những sinh viên có địa chỉ ở Vũng Tàu.
5. Cho biết số tín chỉ của môn Toán rời rạc?
6. Cho biết danh sách các lớp (Tên lớp) thuộc khoa Công nghệ Thông tin?

7. Cho biết danh sách sinh viên (Mã SV, họ tên) của những sinh viên nam có họ “Nguyễn”?
8. Cho biết họ tên, ngày sinh, địa chỉ của những sinh viên nam học lớp 10DHTH1?
9. Cho biết tên những môn học có 1 hoặc 2 tín chỉ?
10. Cho biết họ tên, địa chỉ những sinh viên nữ thuộc khoa Công nghệ Thông tin?
11. Cho biết tên những sinh viên nữ có địa chỉ ở TP. HCM hoặc Vũng Tàu?
12. Cho biết họ tên những sinh viên lớp 10DHTH1 có điểm thi trong khoảng từ 6 đến 8?
13. Cho biết những sinh viên khoa Công nghệ Thông tin học môn CSDL có điểm thi lớn hơn 5?
14. Liệt kê danh sách tên khoa và số lượng lớp trong từng khoa.
15. Liệt kê danh sách tên lớp và số lượng sinh viên trong từng lớp.
16. Liệt kê danh sách tên khoa và số lượng sinh viên trong từng khoa. Sắp xếp các khoa theo thứ tự số lượng sinh viên tăng dần.
17. Liệt kê mã sinh viên, họ tên và điểm trung bình của từng sinh viên.
18. Cho biết họ tên và tổng số môn học của từng sinh viên nam đã học.
19. Cho biết những khoa có ít nhất 2 lớp?
20. Cho biết những lớp có ít nhất 2 sinh viên?
21. Cho biết danh sách các khoa (TENKHOA) có số lượng sinh viên dưới 500 người.
22. Cho biết tên những môn học có số tín chỉ lớn hơn 1 và có ít nhất 2 sinh viên đăng ký học?
23. Cho biết tên những môn học có số tín chỉ lớn nhất và số tín chỉ tương ứng?
24. Kết hợp thông tin sinh viên và thông tin thân nhân thành một tập kết quả (dùng toán tử Union).
25. Cho biết thông tin sinh viên lớp 12DHTP1 (dùng Inner join)?
26. Tìm tất cả những sinh viên không có điểm môn học nào (Dùng Left join)?
27. Cho biết tên sinh viên và mã tất cả các môn học cùng với điểm số của môn học đó mà sinh viên đã tham gia học (tham gia học là có điểm thi) (Dùng right join)?

Bài 2: Sử dụng cơ sở dữ liệu trong bài tập QL_NHANVIEN, thực hiện các yêu cầu sau:

1. Tạo bảng ảo lưu trữ danh sách nhân viên (Mã NV, họ tên), ngày cấp thẻ nhân viên tương ứng với từng nhân viên?
2. Tìm những nhân viên (MANV, HOTEN, NGAYSINH, DIACHI, PHAI, LUONG, MANQL, MAPH) có lương trên 10.000.000 đồng.
3. Cho biết họ tên của những nhân viên nam ở TP. HCM hoặc nhân viên nữ ở Hà Nội.
4. Cho biết mã người quản lý của nhân viên Nguyễn Kim Ánh.

5. Tìm mã và họ tên những nhân viên thuộc phòng Quản trị.
6. Liệt kê danh sách gồm mã, họ tên và địa chỉ của những nhân viên thuộc phòng Nhân sự.
7. Cho biết ngày sinh, địa chỉ và tên phòng làm việc của nhân viên Trần Lê Quyên.
8. Tìm những nhân viên có lương lớn hơn 15.000.000 đồng ở phòng Nhân sự hoặc lương lớn hơn 20.000.000 đồng ở phòng Quản trị. Thông tin bao gồm tất cả các cột trên bảng NHANVIEN.
9. Phòng Quản trị có bao nhiêu nhân viên?
10. Tìm tên những nữ nhân viên và tên người thân của họ.
11. Với mỗi nhân viên cho biết họ tên và số người thân của nhân viên đó.
12. Với mỗi phòng ban liệt kê tên phòng ban và lương trung bình của những nhân viên làm việc cho phòng ban đó.
13. Cho biết danh sách những nhân viên (MANV, HOTEN) có trên hai thân nhân.
14. Cho biết danh sách những nhân viên (MANV, HOTEN) không có thân nhân.
15. Cho biết mã và họ tên nhân viên có lương thấp nhất.
16. Cho biết mã và họ tên nhân viên có lương cao nhất phòng Nhân sự.
17. Cho biết tên những đề án có ít nhất hai nhân viên tham gia.
18. Những nhân viên nào (MANV, HOTEN) tham gia cả hai đề án DA003 và DA004?
19. Những nhân viên nào (HOTEN) có lương lớn hơn lương cao nhất của các nhân viên phòng Quản trị?
20. Phòng ban nào có số nhân viên nhiều hơn số nhân viên của phòng Kế hoạch?
21. Liệt kê ba nhân viên (MANV, HOTEN) có mức lương cao nhất.
22. Cập nhật tăng 10% lương cho những nhân viên đã từng tham gia ít nhất hai đề án.
23. Cho biết mã và họ tên những nhân viên tham gia tất cả các đề án.
24. Có bao nhiêu đề án bắt đầu từ ngày 01/01/2016?
25. Tầng 1 nhà A có bao nhiêu phòng ban?

Bài 3: Sử dụng cơ sở dữ liệu trong bài tập QL_LINHKIEN, thực hiện các yêu cầu sau:

1. Cập nhật tổng tiền cho mỗi hóa đơn theo công thức sau:

$$TONGTIEN = \Sigma (\text{SOLUONG} * \text{DONGIA})$$

2. Cho biết tên những linh kiện được sản xuất bởi nhà sản xuất Genius và có đơn vị tính là Cái.
3. Cho biết thông tin những linh kiện có thời gian bảo hành là 24 tháng.
4. Hóa đơn nào được lập trong tháng 06/2016?
5. Tên và đơn vị tính của các linh kiện có đơn giá lớn hơn 1.000.000 VND.

6. Thông tin những linh kiện (MALK, TENLK, NSX, DVT) được bán ra trước ngày 31/05/2015.
7. Cho biết danh sách những khách hàng mua linh kiện trong tháng 06/2016 có địa chỉ ở TP. HCM.
8. Cho biết tổng số lượng linh kiện trong hóa đơn HD007.
9. Trong tháng 05/2016 có bao nhiêu khách hàng ở Tây Ninh đến mua hàng?
10. Cho biết số điện thoại và địa chỉ của khách hàng có mã KH001.
11. Trong tháng 05/2016 đã lập bao nhiêu đơn hàng?
12. Tổng tiền của hóa đơn HD006 là bao nhiêu?
13. Tổng tiền của 2 hóa đơn HD005 và HD007 là bao nhiêu?
14. Liệt kê mã hóa đơn và số linh kiện khác nhau trong từng hóa đơn.
15. Cho biết tên nhà sản xuất và số linh kiện đã bán của từng nhà sản xuất.
16. Liệt kê mã hóa đơn và tổng tiền của từng hóa đơn.
17. Lập danh sách bao gồm tên khách hàng và mã hóa đơn có tổng tiền lớn hơn 10.000.000 VND.
18. Mỗi loại linh kiện có bao nhiêu linh kiện.
19. Những hóa đơn nào (MAHD) có số linh kiện lớn hơn 10?
20. Cho biết trị giá của những hóa đơn được lập ngày 07/07/2016.
21. Cho biết tên và số lượng của từng mặt hàng trong hóa đơn HD007.

PHẦN 3: BÀI TẬP LÀM THÊM

Bài 1: Sử dụng cơ sở dữ liệu trong bài tập QL_THUVIEN, thực hiện các yêu cầu sau:

1. Cho biết thông tin những cuốn sách thuộc nhà xuất bản Đại học quốc gia TP.HCM.
2. Độc giả Nguyễn Thanh Nam đã mượn bao nhiêu cuốn sách trong ngày 01/06/2016?
3. Những cuốn sách nào của tác giả Nguyễn Hữu Anh?
4. Mỗi loại sách có bao nhiêu cuốn sách?
5. Cho biết mã và tên của những độc giả mượn nhiều sách nhất.
6. Cho biết tên những cuốn sách chưa được mượn lần nào.
7. Những độc giả nào (MADG, TENDG) mượn 3 cuốn sách?

Bài 2: Sử dụng cơ sở dữ liệu trong bài tập QL_THUENHA, thực hiện các yêu cầu sau:

1. Cho biết tiền thuê của căn nhà có địa chỉ 111 Bình Thới, Q11, TP. HCM.
2. Cho biết địa chỉ những căn nhà được xem vào ngày 5/6/2016.
3. Loại nhà cấp 4 ở Quận 12 có bao nhiêu căn?

4. Cho biết thông tin của khách hàng đã xem nhà có địa chỉ 142 Lê Trọng Tấn, Tân Phú, TP. HCM ngày 7/5/2016.
5. Cho biết thông tin loại nhà không có căn nhà nào cho thuê.
6. Cho biết ngày 13/6/2016 có bao nhiêu căn nhà cấp 4 được đăng tin với giá cho thuê thấp hơn 4.000.000 đồng.
7. Cho biết tổng số các căn nhà đã được thuê trong tháng 6/2016 có địa chỉ ở quận 11, TP. HCM.

Bài 3: Sử dụng cơ sở dữ liệu trong bài tập QL_CHUNGCHI, thực hiện các yêu cầu sau:

1. Cho biết ngày bắt đầu và kết thúc của lớp có mã là CCA202.
2. Cho biết thời gian học của lớp CCB101.
3. Lớp CCA101 có bao nhiêu học viên nữ có địa chỉ ở TP.HCM?
4. Những lớp nào có số tiền học phí cao nhất? Thông tin liệt kê gồm: MALH, TENLH, NGAYBD.
5. Cho biết thông tin của những học viên đăng ký học lớp Chứng chỉ A ngày 24/4/2016.
6. Cho biết có bao nhiêu học viên đăng ký lớp chứng chỉ A 2, 4, 6 có ngày bắt đầu là 25/4/2016.
7. Học viên Lê Thành Nam đã đăng ký học mấy lớp chứng chỉ trong năm 2016. Cho biết thông tin của những lớp mà học viên này tham gia: Mã lớp, thời gian học, học phí.



MỤC TIÊU

- Cài đặt các ràng buộc toàn vẹn bằng cấu trúc Constraint.
- Cài đặt các ràng buộc toàn vẹn bằng cấu trúc Trigger.

DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SINH VIÊN

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Máy tính cài đặt SQL Server 2012	1	1	

NỘI DUNG THỰC HÀNH

PHẦN 1: CƠ SỞ LÝ THUYẾT VÀ HƯỚNG DẪN THỰC HÀNH

Trong phần này sử dụng cơ sở dữ liệu QL_NHANVIEN để cài đặt ràng buộc toàn vẹn. Tham khảo chi tiết các lược đồ quan hệ trong cơ sở dữ liệu **QL_NHANVIEN** (*Chương 1. Phần 2. Bài tập 3*).

3.1. Khái niệm

Ràng buộc toàn vẹn (RBT) là những điều kiện bất biến mà tất cả các bộ trong các quan hệ có liên quan (đến ràng buộc) phải thỏa tại mọi thời điểm.

Các đặc trưng của ràng buộc toàn vẹn:

- ✓ Nội dung: Nội dung của RBT là sự mô tả, biểu diễn hình thức của các quy tắc, điều kiện không được vi phạm trong CSDL; có thể được phát biểu bằng ngôn ngữ tự nhiên hoặc ngôn ngữ hình thức (Phép tính quan hệ, Đại số quan hệ, mã giả,...).
- ✓ Bối cảnh: Bối cảnh của một RBT là những quan hệ mà RBT đó có hiệu lực hay nói cách khác đó là những quan hệ cần sử dụng để kiểm tra RBT. Bối cảnh của RBT có thể là một hay nhiều quan hệ.
- ✓ Bảng tầm ảnh hưởng: Khi thực hiện một trong các thao tác cập nhật (thêm, xóa, sửa) trên bối cảnh của một RBT có thể dẫn đến vi phạm RBT. Một thao tác cập nhật trên một quan hệ được xem là hợp lệ nếu nó không vi phạm những RBT có bối cảnh chứa quan hệ đó. Trong quá trình phân tích, thiết kế một cơ sở dữ liệu, người phân tích cần lập bảng tầm ảnh hưởng cho mỗi RBT nhằm xác định thao tác cập nhật nào và cập nhật trên thuộc tính nào của các quan hệ sẽ dẫn đến vi phạm các RBT.

Bảng tầm ảnh hưởng:

Quan hệ	Thêm	Xoá	Sửa
Quan hệ 1	+ / -	+ / -	+ / -
.....			
Quan hệ n	+ / -	+ / -	+ / -

Phân loại ràng buộc toàn vẹn:

- ❖ Ràng buộc toàn vẹn có bối cảnh là một quan hệ
 - Ràng buộc toàn vẹn miền giá trị
 - Ràng buộc toàn vẹn liên bộ
 - Ràng buộc toàn vẹn liên thuộc tính
- ❖ Ràng buộc toàn vẹn có bối cảnh là nhiều quan hệ
 - Ràng buộc toàn vẹn tham chiếu
 - Ràng buộc toàn vẹn liên bộ - liên quan hệ
 - Ràng buộc toàn vẹn liên thuộc tính - liên quan hệ
 - Ràng buộc toàn vẹn do thuộc tính tổng hợp
 - Ràng buộc toàn vẹn do sự hiện diện của chu trình

3.2. Cài đặt ràng buộc toàn vẹn

3.2.1. Cài đặt ràng buộc toàn vẹn dùng Constraint

Cấu trúc Constraint dùng để viết các ràng buộc toàn vẹn đơn giản như ràng buộc khoá chính (*Primary Key*), khoá ngoại (*Foreign Key*), miền giá trị (*Check*), kiểm tra duy nhất (*Unique*), giá trị mặc định (*Default*).

3.2.1.1. Ràng buộc kiểm tra miền giá trị (Check constraint)

Ràng buộc kiểm tra miền giá trị là ràng buộc được dùng để kiểm tra miền giá trị của các cột dữ liệu bên trong bảng. Kiểm tra dữ liệu nhập vào và thông báo lỗi nếu người sử dụng nhập giá trị ngoài phạm vi qui định.

❖ Cách 1: Tạo ràng buộc khi tạo bảng

Ví dụ 3.1: Khi tạo bảng NHANVIEN cần tạo một ràng buộc kiểm tra miền giá trị cho cột lương lớn hơn 0.

```

CREATE TABLE NHANVIEN
(
    MANV CHAR(6) NOT NULL,
    HOTEN NVARCHAR(30),
    NGAYSINH DATE,
    PHAI NCHAR(3),
    DIACHI NVARCHAR(40),
    LUONG MONEY,
    MANQL CHAR(6),
    MAPH CHAR(5)
    PRIMARY KEY (MANV),
    FOREIGN KEY (MAPH) REFERENCES PHONGBAN(MAPH),
    CHECK (LUONG > 0)
)

```

❖ Cách 2: Tạo và đặt tên ràng buộc khi tạo bảng

Cú pháp:

Constraint <Tên_ràng_buộc> Check(Điều kiện)

Bảng NHANVIEN được tạo lại với ràng buộc Check có đặt tên như sau:

```

CREATE TABLE NHANVIEN
(
    MANV CHAR(6) NOT NULL,
    HOTEN NVARCHAR(30),
    NGAYSINH DATE,
    PHAI NCHAR(3),
    DIACHI NVARCHAR(40),
    LUONG MONEY,
    MANQL CHAR(6),
    MAPH CHAR(5)
    PRIMARY KEY (MANV),
    FOREIGN KEY (MAPH) REFERENCES PHONGBAN(MAPH),
    CONSTRAINT CHK_LUONG CHECK (LUONG > 0)
)

```

❖ Cách 3: Tạo và đặt tên ràng buộc sau khi tạo bảng

Cú pháp:

**Alter Table <Tên_bảng>
Add Constraint <Tên_ràng_buộc> Check(Điều kiện)**

Ví dụ 3.2: Sau khi tạo bảng, tạo ràng buộc có tên là CHK_LUONG để kiểm tra cột lương phải lớn hơn 0.

**ALTER TABLE NHANVIEN
ADD CONSTRAINT CHK_LUONG CHECK (LUONG > 0)**

3.2.1.2. Ràng buộc kiểm tra tính duy nhất (Unique constraint)

Là ràng buộc được dùng để kiểm tra tính duy nhất của các cột dữ liệu bên trong bảng. Kiểm tra dữ liệu nhập vào và thông báo lỗi nếu người sử dụng nhập giá trị đã tồn tại.

❖ Cách 1: Tạo ràng buộc khi tạo bảng

Ví dụ 3.3: Khi tạo bảng PHONGBAN cần tạo một ràng buộc kiểm tra tính duy nhất cho cột TENPH.

```
CREATE TABLE PHONGBAN
(
    MAPH CHAR(5) NOT NULL,
    TENPH NVARCHAR(30) UNIQUE,
    DIADIEM NVARCHAR(40),
    PRIMARY KEY (MAPH),
)
```

Hay

```
CREATE TABLE PHONGBAN
(
    MAPH CHAR(5) NOT NULL,
    TENPH NVARCHAR(30),
    DIADIEM NVARCHAR(40),
    PRIMARY KEY (MAPH),
    UNIQUE (TENPH)
)
```

❖ Cách 2: Tạo và đặt tên ràng buộc khi tạo bảng

Cú pháp:

```
Constraint <Tên_ràng_buộc> Unique(<Tên_cột>)
```

Bảng PHONGBAN được tạo lại với ràng buộc Check có đặt tên như sau:

```
CREATE TABLE PHONGBAN
(
    MAPH CHAR(5) NOT NULL,
    TENPH NVARCHAR(30),
    DIADIEM NVARCHAR(40),
    PRIMARY KEY (MAPH),
    CONSTRAINT UNI_TENPH UNIQUE (TENPH)
)
```

❖ Cách 3: Tạo và đặt tên ràng buộc sau khi tạo bảng

Cú pháp:

```
Alter Table <Tên_bảng>
Add Constraint <Tên_ràng_buộc> Unique(<Tên_cột>)
```

Ví dụ 3.4: Sau khi tạo bảng, tạo ràng buộc có tên là UNI_TENPH kiểm tra tính duy nhất cho cột TENPH.

```
ALTER TABLE PHONGBAN  
ADD CONSTRAINT UNI_TENPH UNIQUE (TENPH)
```

3.2.1.3. Ràng buộc kiểm tra giá trị mặc định (Default constraint)

Là ràng buộc trả về giá trị mặc định khi để trống cột có định nghĩa ràng buộc này. Nếu người sử dụng không nhập giá trị thì mặc định chọn giá trị đã được thiết lập trong ràng buộc.

❖ Cách 1: Tạo ràng buộc khi tạo bảng

Ví dụ 3.5: Khi tạo bảng PHONGBAN cần tạo một ràng buộc giá trị mặc định thực hiện điền vào giá trị “Chưa xác định” khi thêm dữ liệu để trống cột DIADIEM.

```
CREATE TABLE PHONGBAN  
(  
    MAPH CHAR(5) NOT NULL,  
    TENPH NVARCHAR(30),  
    DIADIEM NVARCHAR(40) DEFAULT 'CHUA XAC DINH',  
    PRIMARY KEY (MAPH),  
)
```

❖ Cách 2: Tạo và đặt tên ràng buộc sau khi tạo bảng

Cú pháp:

```
Alter Table <Tên_bảng>  
Add Constraint <Tên_ràng_buộc> Default <Giá trị>  
For <tên_cột>
```

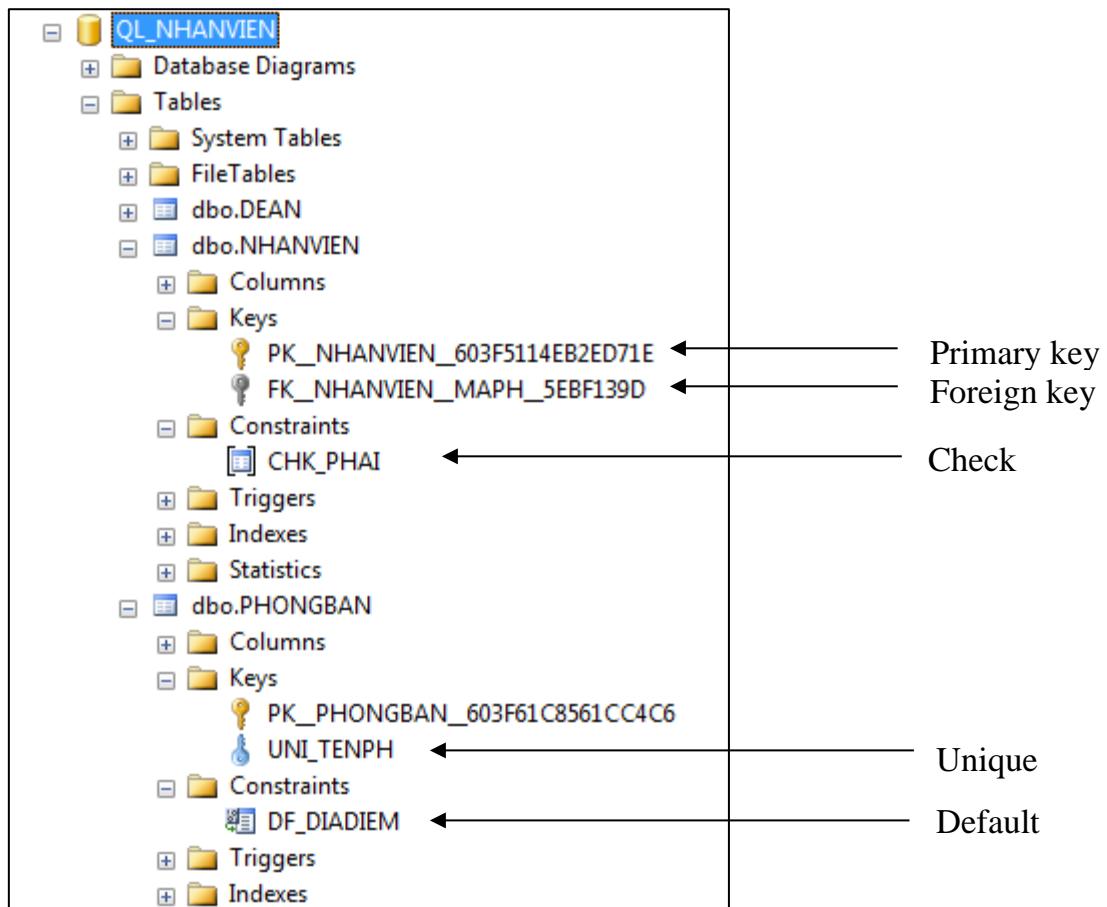
Ví dụ 3.6: Sau khi tạo bảng, tạo ràng buộc có tên là DF_DIADIEM với giá trị mặc định là “Chưa xác định” áp dụng cho cột DIADIEM.

```
ALTER TABLE PHONGBAN  
ADD CONSTRAINT DF_DIADIEM DEFAULT 'CHUA XAC DINH' FOR DIADIEM
```

3.2.1.4. Quản lý các ràng buộc Constraint

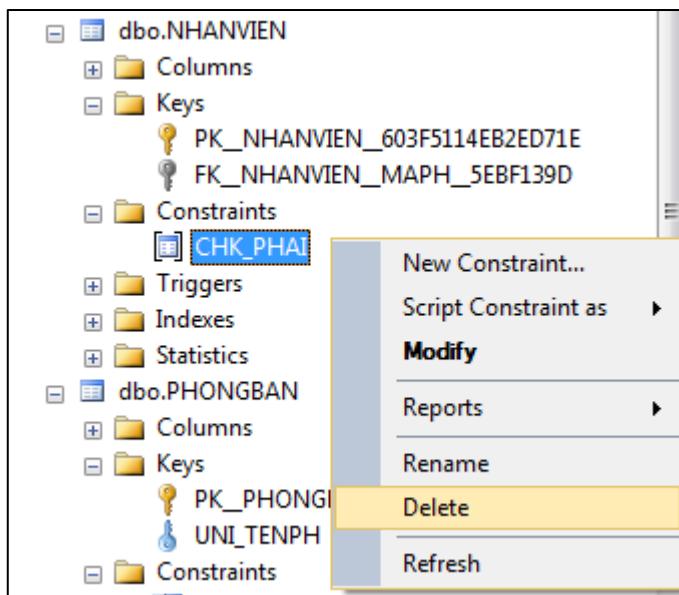
❖ Xem ràng buộc

Mở rộng danh mục bảng đã tạo các ràng buộc trên đó, trong thư mục *Keys* chứa các ràng buộc *Primary Key*, *Foreign Key*, *Unique*. Trong thư mục *Constraints* chứa các ràng buộc *Check*, *Default*.



❖ Xoá ràng buộc

- ✓ **Cách 1:** Mở thư mục chứa ràng buộc → nhấn chuột phải vào ràng buộc cần xoá → chọn *Delete*.



- ✓ **Cách 2:** dùng lệnh để xoá ràng buộc theo cú pháp

```
Alter table <Tên_bảng>
Drop Constraint <Định danh các ràng buộc>
```

Ví dụ 3.7: Xoá ràng buộc có tên là CHK_PHAI

```
ALTER TABLE NHANVIEN  
DROP CONSTRAINT CHK_PHAI
```

3.2.2. Cài đặt ràng buộc toàn vẹn dùng Trigger

Để xây dựng và thiết kế một cơ sở dữ liệu đáp ứng được các yêu cầu của người sử dụng, có tính ứng dụng thực tế cao đòi hỏi cơ sở dữ liệu đó phải lưu trữ thông tin một cách hợp lý, nhất quán, đầy đủ và chính xác. Để làm được điều này T-SQL cung cấp một cấu trúc lệnh đặc biệt làm cho việc lập trình CSDL trở nên mạnh mẽ và uyển chuyển hơn đó chính là TRIGGER. Trong SQL Server 2012 có hai dạng trigger là DML trigger và DDL trigger. Trong tài liệu này chỉ giới thiệu DML trigger nhằm mục đích phục vụ cho việc cài đặt ràng buộc toàn vẹn để kiểm tra dữ liệu trên bảng.

3.2.2.1. Khái niệm

Trigger là một cấu trúc lệnh T-SQL đặc biệt có khả năng thực thi một cách tự động mỗi khi cập nhật (thêm, xoá, sửa) dữ liệu trên bảng liên quan với trigger.

Cấu trúc lệnh trigger được dùng để kiểm tra sự toàn vẹn dữ liệu của tất cả các ràng buộc toàn vẹn có trong cơ sở dữ liệu. Một số ràng buộc đơn giản như liên bộ (khoá chính), tham chiếu (khoá ngoại), miền giá trị thì chỉ cần sử dụng cấu trúc Constraint. Sử dụng trigger nếu ràng buộc phức tạp hoặc khi cần sử dụng tính tự động của trigger để thực hiện một số công việc khác.

3.2.2.2. Cấu trúc tổng quát

```
CREATE TRIGGER <Tên_trigger> ON <Tên_bảng>  
[WITH ENCRYPTION]  
FOR <Hành động kích hoạt>  
AS  
<Các lệnh T-SQL>
```

Trong đó:

- ✓ Tên_trigger: là tên của trigger cần tạo, tên này phải duy nhất trong một CSDL.
- ✓ ON: cho biết trigger đang viết cho bảng dữ liệu nào.
- ✓ Tên_bảng: là tên của bảng dữ liệu cần tạo trigger trên đó.
- ✓ WITH ENCRYPTION: dùng để mã hoá nội dung trigger.
- ✓ Hành động kích hoạt có 3 hành động: INSERT, DELETE, UPDATE.
- ✓ Các lệnh T-SQL: là các lệnh SQL thực hiện các xử lý trong trigger. Một số lệnh sau đây không được sử dụng trong trigger: ALTER DATABASE, CREATE DATABASE, DISK INIT, DISK RESIZE, DROP DATABASE, LOAD DATABASE, LOAD LOG, RECONFIGURE, RESTORE DATABASE, RESTORE LOG.

3.2.2.3. Phân loại trigger

Tùy thuộc vào mục đích kiểm tra ràng buộc toàn vẹn, có thể lựa chọn một trong hai loại trigger sau:

- ✓ **After trigger:** Xảy ra sau khi thực hiện hành động thêm (*Insert*), xoá (*Delete*), sửa (*Update*) dữ liệu.
- ✓ **Instead of trigger:** Xảy ra khi thực hiện hành động thêm (*Insert*), xoá (*Delete*), sửa (*Update*) dữ liệu.

Ứng với mỗi loại trigger trên với các thao tác thêm, xoá, sửa dữ liệu có thể lựa chọn hành động kích hoạt tương ứng *Insert*, *Delete*, *Update*.

❖ Insert trigger

Insert trigger được kích hoạt khi xảy ra hành động thêm dữ liệu vào bảng. Hệ thống sẽ phát sinh một bảng phụ *INSERTED* để chứa một dòng dữ liệu mới cần thêm vào bảng. Bảng phụ này có cấu trúc hoàn toàn giống với bảng được định nghĩa trên trigger đó. Sau khi trigger kết thúc hoạt động thì bảng phụ này sẽ bị huỷ bỏ.

Ví dụ 3.8: Một RBTV được phát biểu như sau “Lương của nhân viên phải lớn hơn 0”.

Ràng buộc này có các đặc trưng:

Nội dung: $\forall \text{ nv} \in \text{NHANVIEN}, \text{nv.LUONG} > 0$

Bối cảnh: NHANVIEN

Bảng tầm ảnh hưởng:

	Thêm	Xoá	Sửa
NHANVIEN	+	-	+ (LUONG)

Yêu cầu: Viết trigger kiểm tra khi nhập dữ liệu vào bảng NHANVIEN thì LUONG lớn hơn 0.

```
CREATE TRIGGER KT_LUONG ON NHANVIEN
FOR INSERT
AS
IF(SELECT LUONG FROM INSERTED) > 0
    COMMIT TRAN
ELSE
    BEGIN
        PRINT 'LUONG PHAI LON HON 0!'
        ROLLBACK TRAN
    END
```

Khi thêm một dòng dữ liệu vào bảng NHANVIEN với lương không hợp lệ thì sẽ xuất hiện như hình dưới đây:

```

INSERT INTO NHANVIEN
VALUES ('NV0010', N'Trần Minh Tuấn', '17/09/1980', N'Nam', N'TP. HCM', -18000000, NULL, 'PH002');

100 % < 
Messages
LUONG PHAI LON HON 0!
Msg 3609, Level 16, State 1, Line 1
The transaction ended in the trigger. The batch has been aborted.

```

Lưu ý: Thông báo “LUONG PHAI LON HON 0 !” là do thiết lập trong cú pháp tạo trigger, còn thông báo 3609 “The transaction ended in the trigger. The batch has been aborted” là do sử dụng lệnh Rollback trong thân trigger. Những trigger không sử dụng lệnh Rollback thì lỗi này sẽ không xảy ra. Tuy nhiên, lỗi này chỉ thông báo cho biết giao tác đã kết thúc và bỏ lệnh thực hiện đã được hủy, điều này hoàn toàn không gây ảnh hưởng đến hoạt động của trigger.

Trong ví dụ trên chúng ta sử dụng mệnh đề *For Insert* để chỉ ra rằng trigger này chỉ hoạt động cho hành động thêm dữ liệu vào bảng NHANVIEN. Lệnh *IF (SELECT LUONG FROM INSERTED) > 0* sẽ kiểm tra nếu lương trong bảng tạm Inserted lớn hơn 0 thì thực hiện giao tác bởi lệnh Commit Tran, nghĩa là dữ liệu hợp lệ và dữ liệu được thêm vào bảng. Ngược lại thì quay lui giao tác bởi lệnh Rollback Tran, khi đó dữ liệu không được thêm vào bảng.

❖ Delete trigger

Delete trigger được kích hoạt khi xảy ra hành động xoá dữ liệu ra khỏi bảng. Hệ thống sẽ phát sinh một bảng phụ DELETED để chứa một dòng dữ liệu muốn xoá ra khỏi bảng. Bảng phụ này có cấu trúc hoàn toàn giống với bảng được định nghĩa trên trigger đó. Sau khi trigger kết thúc hoạt động thì bảng phụ này sẽ bị huỷ bỏ.

Ví dụ 3.9: Viết trigger kiểm tra chỉ cho xoá những nhân viên sinh trước năm 1966.

```

CREATE TRIGGER XOA_NV ON NHANVIEN
FOR DELETE
AS
IF( SELECT YEAR(NGAYSINH) FROM DELETED ) < 1966
    COMMIT TRAN
ELSE
    BEGIN
        PRINT 'KHONG THE XOA SINH NHAN VIEN NAY!'
        ROLLBACK TRAN
    END

```

Khi xoá những nhân viên sinh từ năm 1966 về sau khỏi bảng NHANVIEN thì sẽ xuất hiện như hình dưới đây:

```

DELETE FROM NHANVIEN
WHERE YEAR(NGAYSINH) = 1966

```

100 %

Messages

KHONG THE XOA SINH NHAN VIEN NAY!

Msg 3609, Level 16, State 1, Line 1

The transaction ended in the trigger. The batch has been aborted.

❖ Update trigger

Update trigger được kích hoạt khi xảy ra hành động sửa dữ liệu trên bảng. Hệ thống sẽ phát sinh hai bảng phụ, *INSERTED* để chứa dòng dữ liệu mới cần thêm vào bảng và *DELETED* để chứa dòng dữ liệu muốn xoá ra khỏi bảng. Sau khi trigger kết thúc hoạt động thì hai bảng phụ này sẽ bị huỷ bỏ.

Ví dụ 3.10: Viết trigger chỉ cho cập nhật tăng lương của nhân viên không vượt quá 10%.

```

CREATE TRIGGER KT_CN_LUONG ON NHANVIEN
FOR UPDATE
AS
IF (SELECT LUONG FROM INSERTED) <= 1.1 * (SELECT LUONG FROM DELETED)
    COMMIT TRAN
ELSE
    BEGIN
        PRINT 'TANG LUONG KHONG QUA 10%!'
        ROLLBACK TRAN
    END

```

Khi tăng lương của nhân viên có mã NV0001 trong bảng NHANVIEN, nếu tăng 20% thì sẽ xuất hiện như hình dưới đây:

```

UPDATE NHANVIEN
SET LUONG = LUONG * 1.2
WHERE MANV = 'NV0001';

```

100 %

Messages

TANG LUONG KHONG QUA 10%

Msg 3609, Level 16, State 1, Line 1

The transaction ended in the trigger. The batch has been aborted.

❖ Trigger nhiều hành động

Trong một trigger có thể khai báo nhiều hành động insert, delete, update thay vì viết ba trigger để khai báo mỗi hành động tương ứng. Trigger dạng này sẽ có không gian hoạt động rộng hơn và hiệu quả hơn.

Ví dụ 3.11: Viết trigger thêm hay sửa lương của nhân viên thì lương phải lớn hơn 0.

```
CREATE TRIGGER KT_LUONG ON NHANVIEN
FOR INSERT, UPDATE
AS
IF(SELECT LUONG FROM INSERTED) > 0
    COMMIT TRAN
ELSE
BEGIN
    PRINT 'LUONG PHAI LON HON 0!'
    ROLLBACK TRAN
END
```

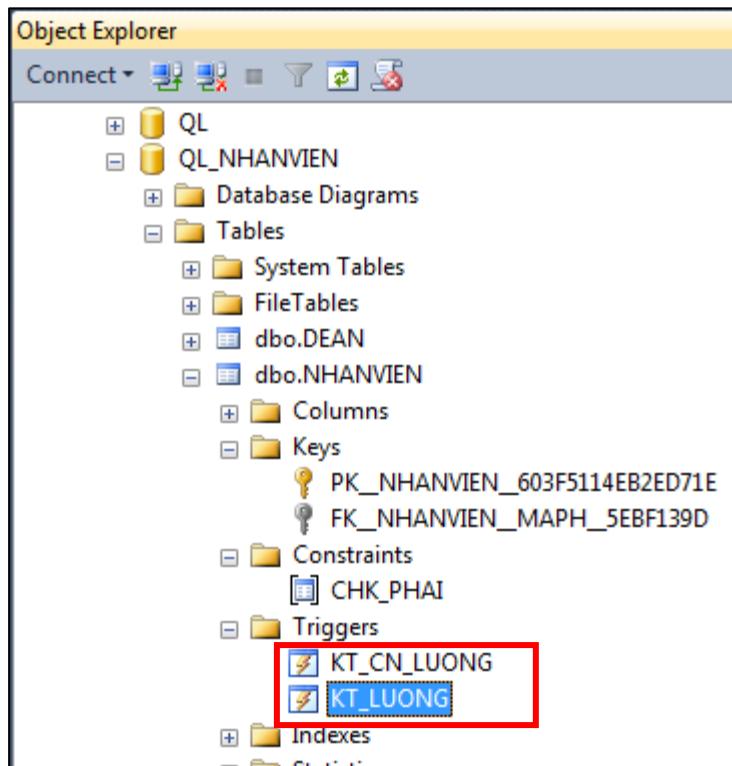
Trigger này sẽ được kích hoạt với một trong hai hành động là insert hoặc update.

3.2.2.4. Quản lý trigger

Các trigger được tạo trên các bảng trong cơ sở dữ liệu cần phải được quản lý.

❖ Lưu trữ trigger

Tất cả các trigger có trên bảng được lưu trữ trong mục Triggers của bảng đó như hình sau:



Hình 3.1. Giao diện quản lý Trigger

Ngoài ra, có thể xem các trigger có trên bảng bằng thủ tục hệ thống:

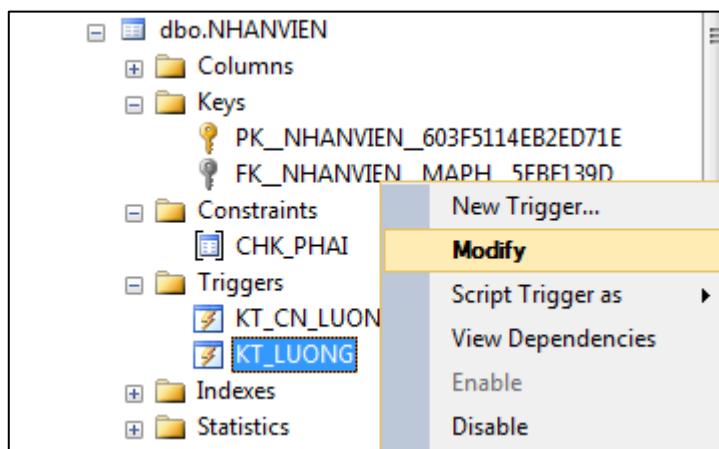
```
sp_helptrigger <Tên_bảng>
```

Ví dụ 3.12: Xem các trigger trên bảng NHANVIEN

	trigger_name	trigger_owner	isupdate	isdelete	isinsert	isafter	isinsteadof	trigger_schema
1	KT_LUONG	dbo	0	0	1	1	0	dbo
2	KT_CN_LUONG	dbo	1	0	0	1	0	dbo
3	XOA_NV	dbo	0	1	0	1	0	dbo

❖ Xem nội dung trigger:

- ✓ **Cách 1:** từ nơi lưu trữ trigger nhấp chuột phải vào tên trigger cần xem nội dung > chọn Modify.



Hình 3.2. Giao diện xem nội dung Trigger

- ✓ **Cách 2:** xem nội dung của trigger bằng cách dùng thủ tục hệ thống

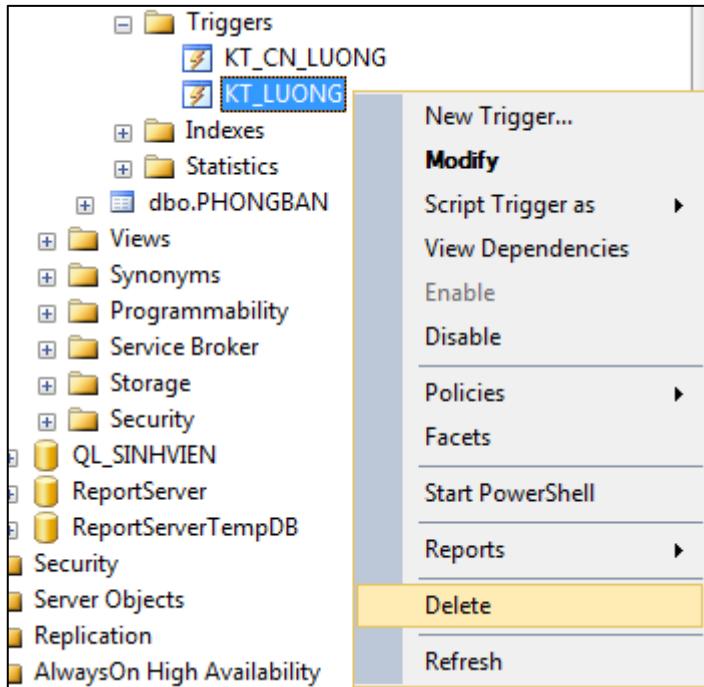
```
sp_helptext <Tên_trigger>
```

Ví dụ 3.13: Xem nội dung trigger có tên KT_LUONG.

Text
1 CREATE TRIGGER KT_LUONG ON NHANVIEN
2 FOR INSERT
3 AS
4 IF(SELECT LUONG FROM INSERTED) > 0
5 COMMIT TRAN
6 ELSE
7 BEGIN
8 PRINT 'LUONG PHAI LON HON 0!'
9 ROLLBACK TRAN
10 END

❖ Xoá trigger

- ✓ **Cách 1:** để xoá trigger thực hiện nhấn chuột phải vào trigger cần xoá và chọn Delete.

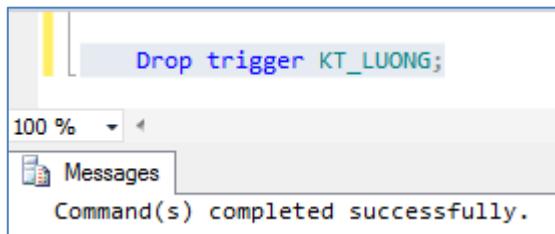


Hình 3. 3. Giao diện xoá Trigger

- ✓ **Cách 2:** dùng lệnh để xóa trigger

```
drop trigger <Tên_trigger>
```

Ví dụ 3.14: Xoá trigger có tên là KT_LUONG.



PHẦN 2: BÀI TẬP THỰC HÀNH

Bài 1: Sử dụng cơ sở dữ liệu trong bài tập QL_SINHVIEN, thực hiện cài đặt các ràng buộc toàn vẹn sau:

1. Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
2. Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho các cột TENKH, TENLOP, TENMH.
3. Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: LOPTRUONG có giá trị mặc định là *Chưa xác định*, SOTC mặc định là 3.
4. Viết trigger kiểm tra mỗi sinh viên chỉ được thi tối đa 3 lần cho mỗi môn học.

- Viết trigger thực hiện cập nhật số SISO trên bảng LOP mỗi khi thêm, xóa hay sửa dữ liệu trên bảng SINHVIEN.

Bài 2: Sử dụng cơ sở dữ liệu trong bài tập QL_NHANVIEN, thực hiện cài đặt các ràng buộc toàn vẹn sau:

- Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
- Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho các cột TENPH, TENDA.
- Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: QUANHE có giá trị mặc định là *Không xác định*, LUONG mặc định là 8.000.000.
- Viết trigger thực hiện kiểm tra khi phân công đề án cho nhân viên thì ngày tham gia (THOIGIAN) phải lớn hơn hoặc bằng ngày bắt đầu đề án (NGAYBD).
- Viết trigger thực hiện kiểm tra khi thêm một nhân viên thì tuổi của nhân viên phải lớn hơn hoặc bằng 18.

Bài 3: Sử dụng cơ sở dữ liệu trong bài tập QL_LINHKIEN, thực hiện cài đặt các ràng buộc toàn vẹn sau:

- Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
- Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho các cột TENLK, TENLOAI.
- Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: DIACHI có giá trị mặc định là *Không xác định*, TGBH mặc định là 12 tháng.
- Viết trigger thực hiện kiểm tra khi thêm một linh kiện thì ngày sản xuất phải bé hơn ngày hiện tại.
- Viết trigger thực hiện cập nhật tổng tiền trên bảng HOADON mỗi khi thêm hay sửa một chi tiết hóa đơn trong bảng CHITIETHD.

$$\text{Tổng tiền} = \sum (\text{Số lượng} * \text{Đơn giá}) \text{ của cùng số hóa đơn.}$$

PHẦN 3: BÀI TẬP LÀM THÊM

Bài 1: Sử dụng cơ sở dữ liệu trong bài tập QL_THUVIEN, thực hiện cài đặt các ràng buộc toàn vẹn sau:

- Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
- Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho các cột TENNXB, TENLOAI.
- Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: NGAYTRA có giá trị mặc định là *NULL*.

4. Viết trigger thực hiện kiểm tra khi một độc giả mượn sách thì ngày mượn phải bé hơn hoặc bằng ngày trả.
5. Viết trigger thực hiện kiểm tra khi thêm hay sửa thông tin một độc giả thì tuổi của độc giả phải lớn hơn hoặc bằng 18.

Bài 2: Sử dụng cơ sở dữ liệu trong bài tập QL_THUENHA , thực hiện cài đặt các ràng buộc toàn vẹn sau:

1. Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
2. Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho cột TENLN.
3. Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: GHICHU có giá trị mặc định là CHUA THUÊ.
4. Viết trigger thực hiện kiểm tra ngày đăng tin phải lớn hơn hoặc bằng 7.
5. Viết trigger thực hiện kiểm tra khi một khách hàng xem nhà, ngày xem nhà phải sau ngày đăng tin.

Bài 3: Sử dụng cơ sở dữ liệu trong bài tập QL_CHUNGCHI, thực hiện cài đặt các ràng buộc toàn vẹn sau:

1. Dùng Check Constraint viết tất cả các ràng buộc miền giá trị có trong cơ sở dữ liệu trên.
2. Dùng Unique Constraint viết ràng buộc kiểm tra duy nhất cho các cột TENLCC, TENCC.
3. Dùng Default Constraint viết các ràng buộc giá trị mặc định cho các cột như sau: TIENCONLAI có giá trị mặc định là 0.
4. Dùng Check Constraint viết các ràng buộc cho cột THOIGIANHOC có giá trị là 2, 4, 6 hoặc 3, 5, 7.
5. Viết trigger kiểm tra khi thêm vào một lớp học thì học phí phải lớn hơn 0.
6. Viết trigger thực hiện kiểm tra khi một học viên đăng ký học thì tiền đặt cọc phải lớn hơn hoặc bằng 50% tiền học phí.

TÀI LIỆU THAM KHẢO

- [1]. Hoàng Thị Liên Chi, Nguyễn Văn Lẽ (2013), Giáo trình cơ sở dữ liệu, Trường Đại học Công nghiệp Thực phẩm TP. HCM, 2013.
- [2]. Phạm Hữu Khang (2007), *SQL Server 2005-Lập trình T-SQL*, NXB Lao động Xã hội.
- [3]. Phạm Hữu Khang, Đoàn Thiện Ngân, *SQL Server 2005 - Lập trình thủ tục và hàm*, Nhà xuất bản Lao động Xã hội, 2008.
- [4]. Hồ Thuần, Hồ Cẩm Hà (2007), *Các hệ cơ sở dữ liệu Lý thuyết và thực hành*, Nhà xuất bản Giáo dục.
- [5]. Nguyễn Bá Tường (2005), *Cơ sở dữ liệu Lý thuyết và thực hành*, Nhà xuất bản Khoa học và Kỹ thuật.
- [6]. Jeffrey D.Ullman (1989), *The Principles of Database and Knowledge Base System Vol1*, 2, Computer Science Press.
- [7]. Ramez Elmasri, Shamkant B. Navathe (1997), *Fundamentals of Database Systems*, International Editor.
- [8]. Books Online for SQL Server 2012.
- [9]. Tutorials for SQL Server 2012.

PHỤ LỤC

HƯỚNG DẪN CÀI ĐẶT SQL SERVER 2012

Trước khi cài đặt SQL Server 2012 cần phải chắc chắn rằng cấu hình máy tính đáp ứng được các yêu cầu đối với SQL Server 2012. Có rất nhiều các yêu cầu khác nhau về SQL Server 2012 mà máy tính phải đáp ứng, nó phụ thuộc vào phiên bản SQL Server cần cài đặt và nền tảng của hệ điều hành. Có thể kiểm tra thông tin cài đặt trên trang web: <http://msdn.microsoft.com/en-us/library/ms143506.aspx>

Lưu ý: Một số yêu cầu tối thiểu để cài đặt SQL Server 2012 phiên bản Developer
Yêu cầu về phần cứng:

- Processor: Pentium IV 2Gb MHz trở lên.
- Ram: 1GB trở lên.
- Hard Disk: 10GB.

Yêu cầu về phần mềm:

- Hệ điều hành tối thiểu: Windows XP SP3, Windows Vista SP1, Windows 7, Windows 2003 SP2.
- Net Framework 4.0
- Windows Installer 4.5 trở lên.

Các bước thực hiện cài đặt SQL Server 2012 Developer

Bước 1: Cài đặt Windows Installer 4.5

Nếu hệ thống chưa cài đặt Windows Installer 4.5 thì tiến hành cài đặt Windows Installer 4.5 hoặc cao hơn.

Bước 2: Cài đặt SQL Server

Đầu tiên tải bản SQL Server Express 2012 miễn phí tại trang chủ Microsoft theo địa chỉ <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Lưu ý: Nhớ để máy tính kết nối Internet trong quá trình cài đặt.

Microsoft

Download Center

Shop ▾ Products ▾ Categories ▾ Support ▾ Security ▾

 Microsoft® SQL Server® 2012 Express

Select Language: English ▼

Download

Microsoft® SQL Server® 2012 Express is a powerful and reliable free data management system that delivers a rich and reliable data store for lightweight Web Sites and desktop applications.

+ Details
+ System Requirements
+ Install Instructions

Chọn ENU\64\SQLEXPRESS_x64_ENU.exe nếu máy tính cài Windows 64 bit.
 Ngược lại chọn ENU\86\SQLEXPRESS_x86_ENU.exe nếu máy tính cài Windows 32 bit.

Choose the download you want

File Name	Size
ENU\64\SQLEXPRESS_x64_ENU.exe	132.3 MB
ENU\64\SQLEXPRESSADV_x64_ENU.exe	1.3 GB
ENU\64\SQLEXPRESSW_x64_ENU.exe	669.9 MB
ENU\x64\SqlLocalDB.MSI	33.0 MB
ENU\x64\SQLManagementStudio_x64_ENU.exe	600.2 MB
ENU\x86\SQLEXPRESS_x86_ENU.exe	116.7 MB

Download Summary:
 You have not selected any file(s) to download.

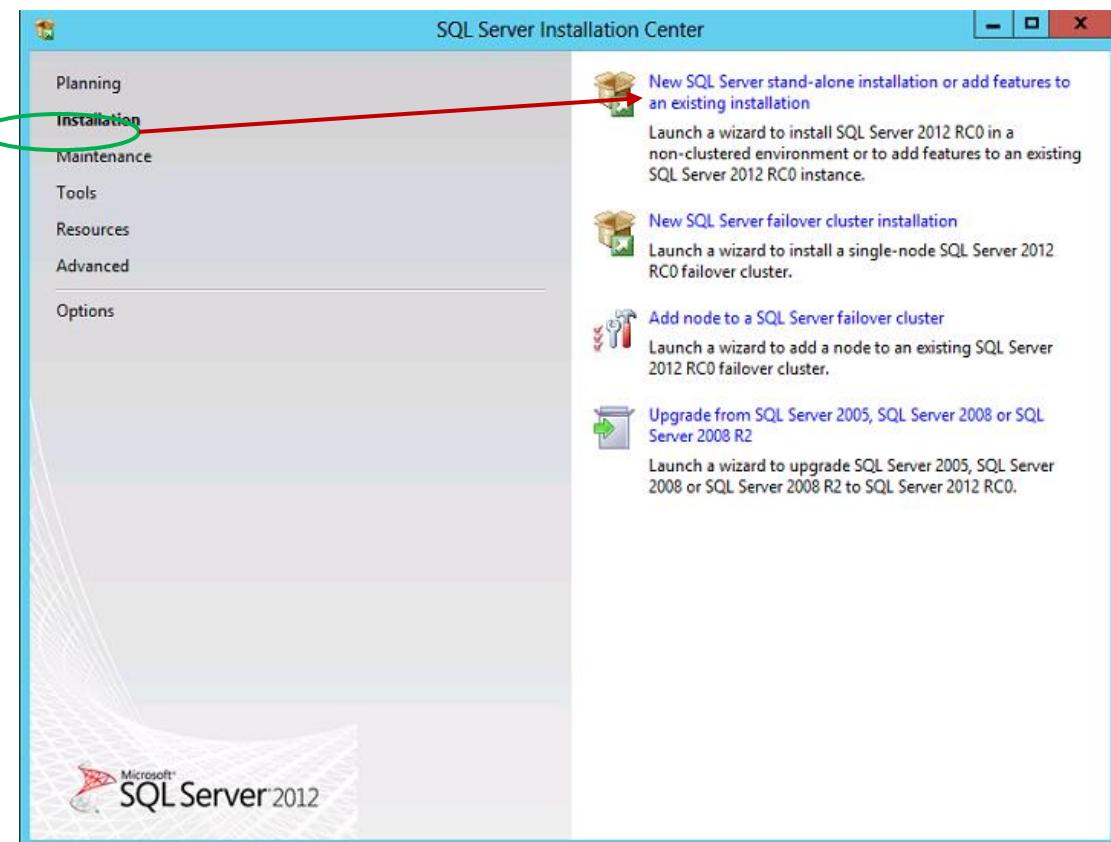
Total Size: 0

Next

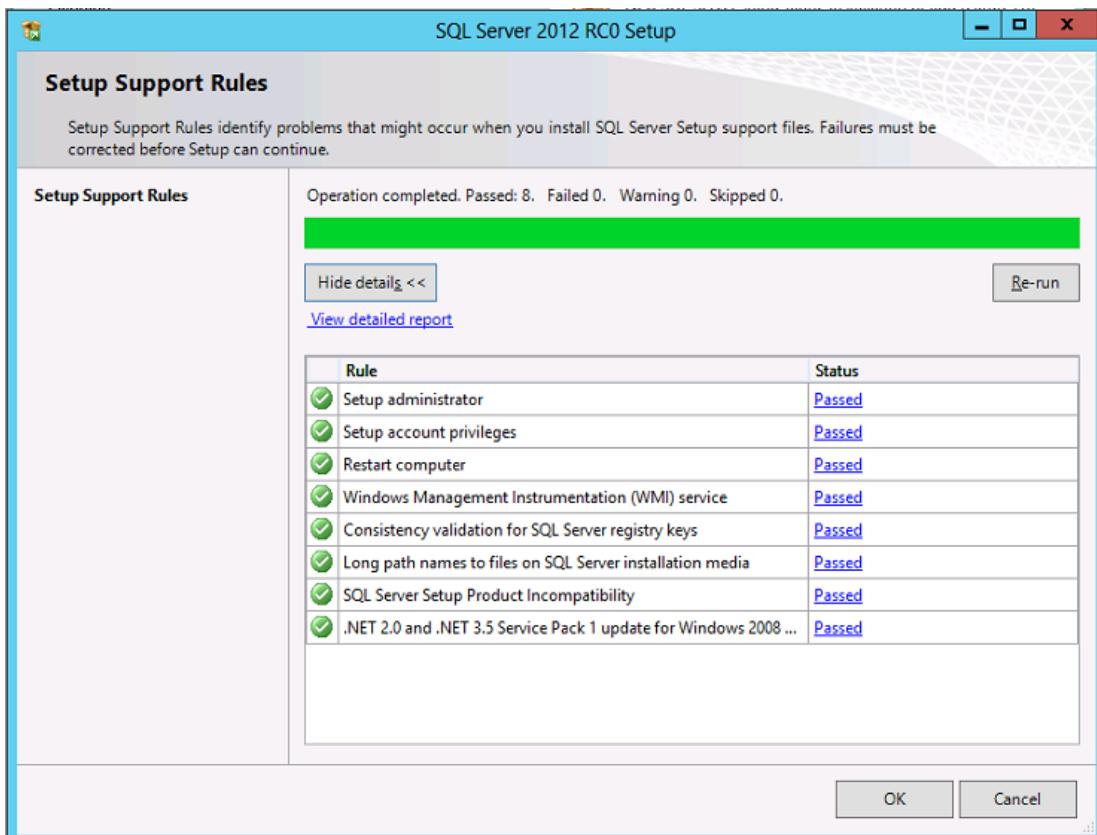
Sau khi download về mở file setup.exe lên.

1033_ENU_LP	11/10/2011 7:03 PM	File folder
redist	11/10/2011 7:04 PM	File folder
resources	11/10/2011 7:03 PM	File folder
StreamInsight	11/10/2011 7:03 PM	File folder
Tools	11/10/2011 7:03 PM	File folder
x64	11/10/2011 7:04 PM	File folder
x86	11/10/2011 7:05 PM	File folder
autorun	10/29/2011 7:41 AM	Setup Information 1 KB
MediaInfo	11/6/2011 1:05 PM	XML Document 1 KB
setup	11/5/2011 9:10 AM	Application 187 KB
setup.exe.config	10/29/2011 6:41 AM	CONFIG File 1 KB
sqmapi.dll	11/5/2011 9:00 AM	Application extens... 137 KB

Trong cửa sổ cài đặt chọn Installation, sau đó chọn New SQL Server stand-alone installation.

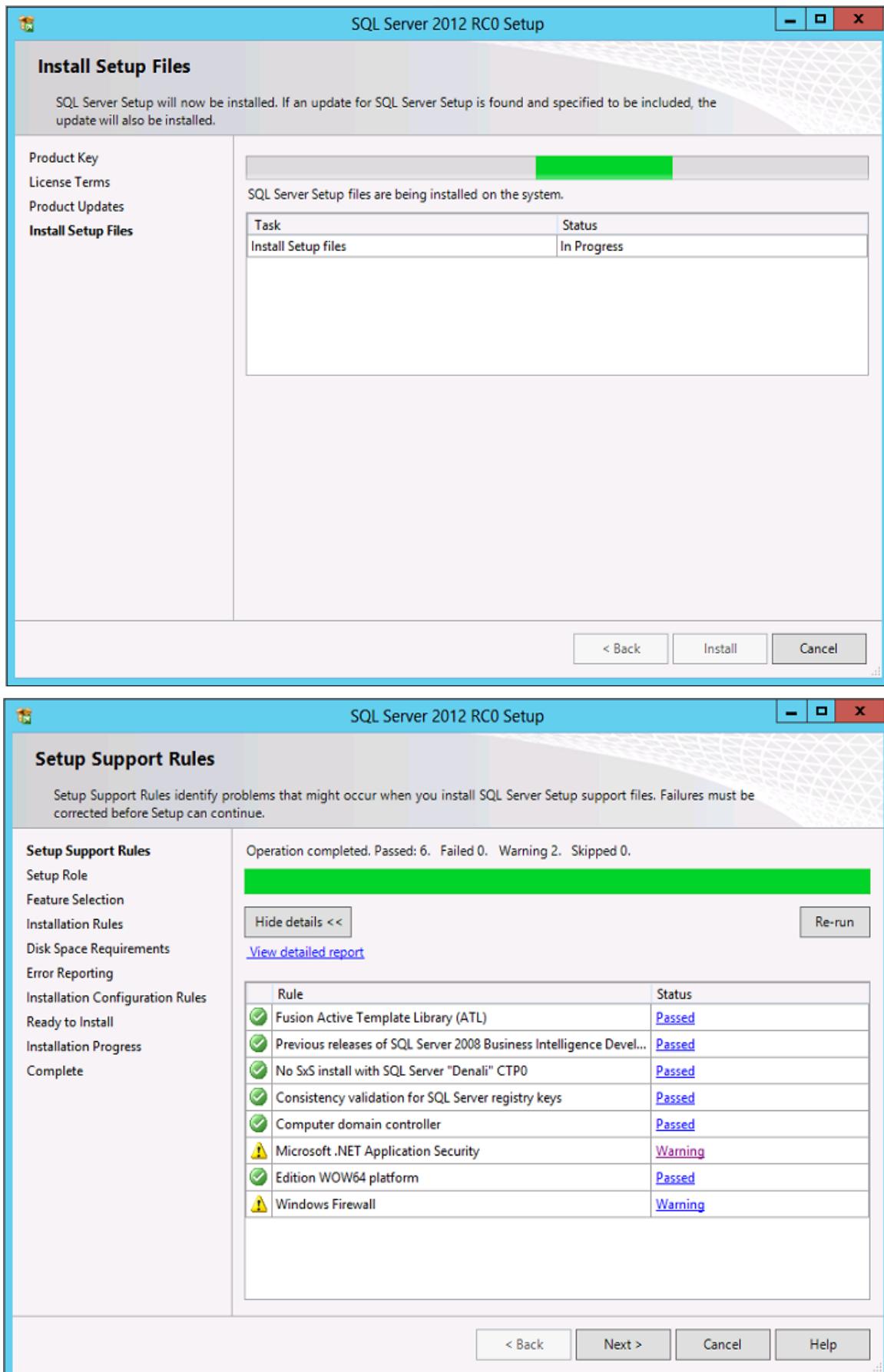


Xuất hiện hộp thoại thông báo quá trình cài đặt như sau:

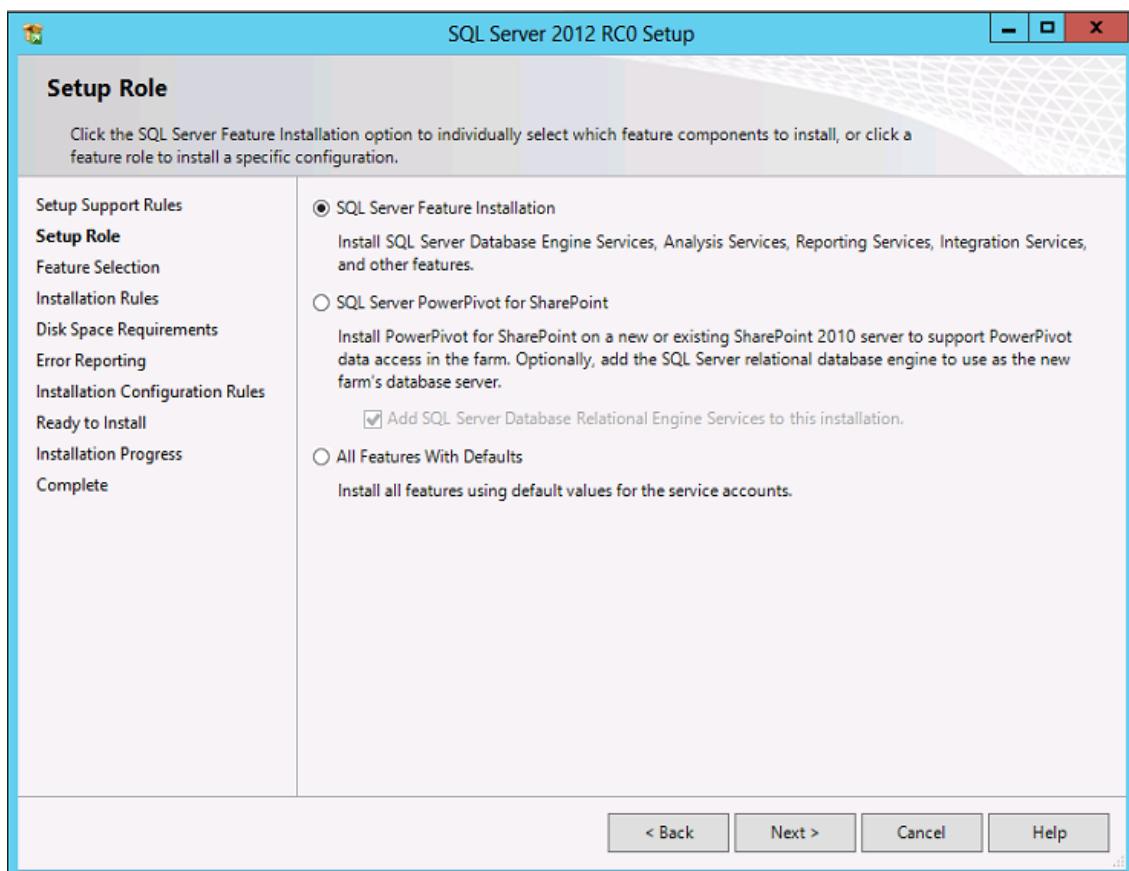


Hình 1.1. Giao diện SQL Server Setup

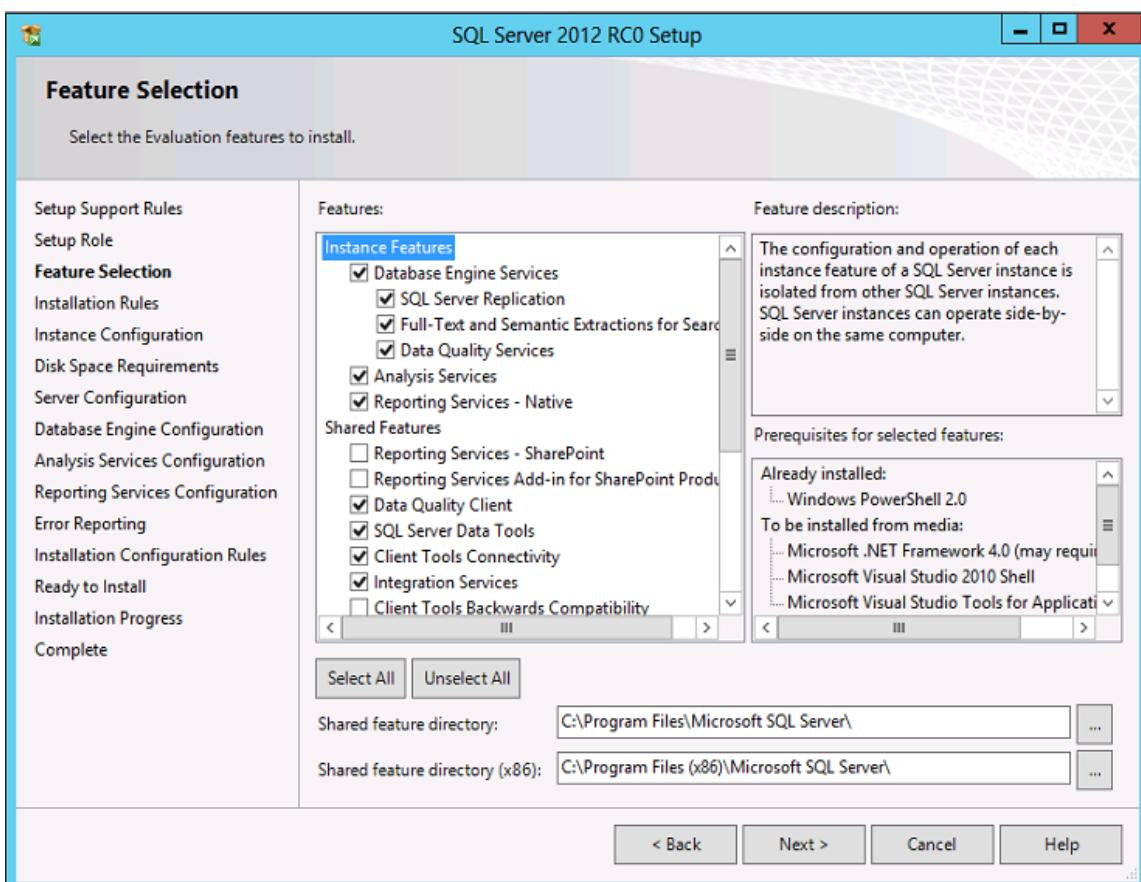
Chọn OK → bắt đầu quá trình cài đặt như sau:



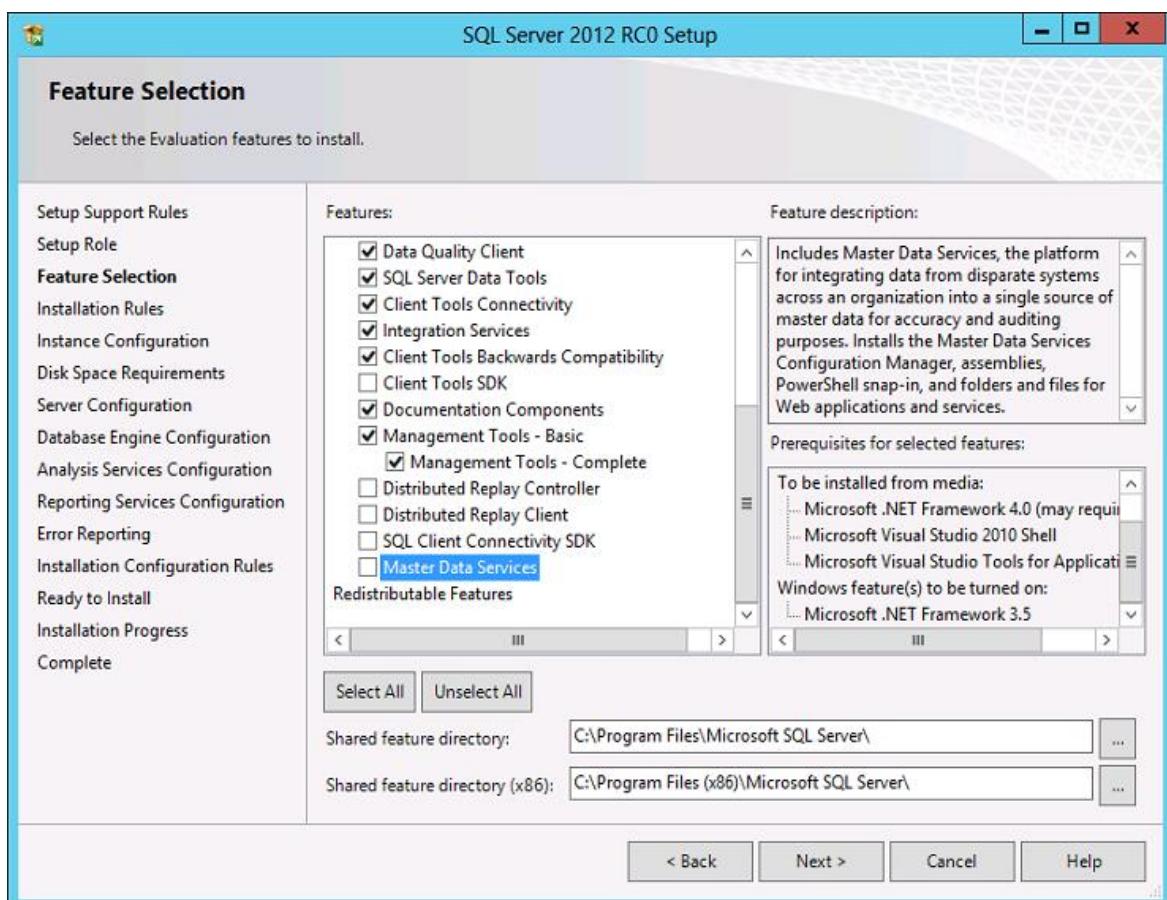
Tiếp tục chọn Next.



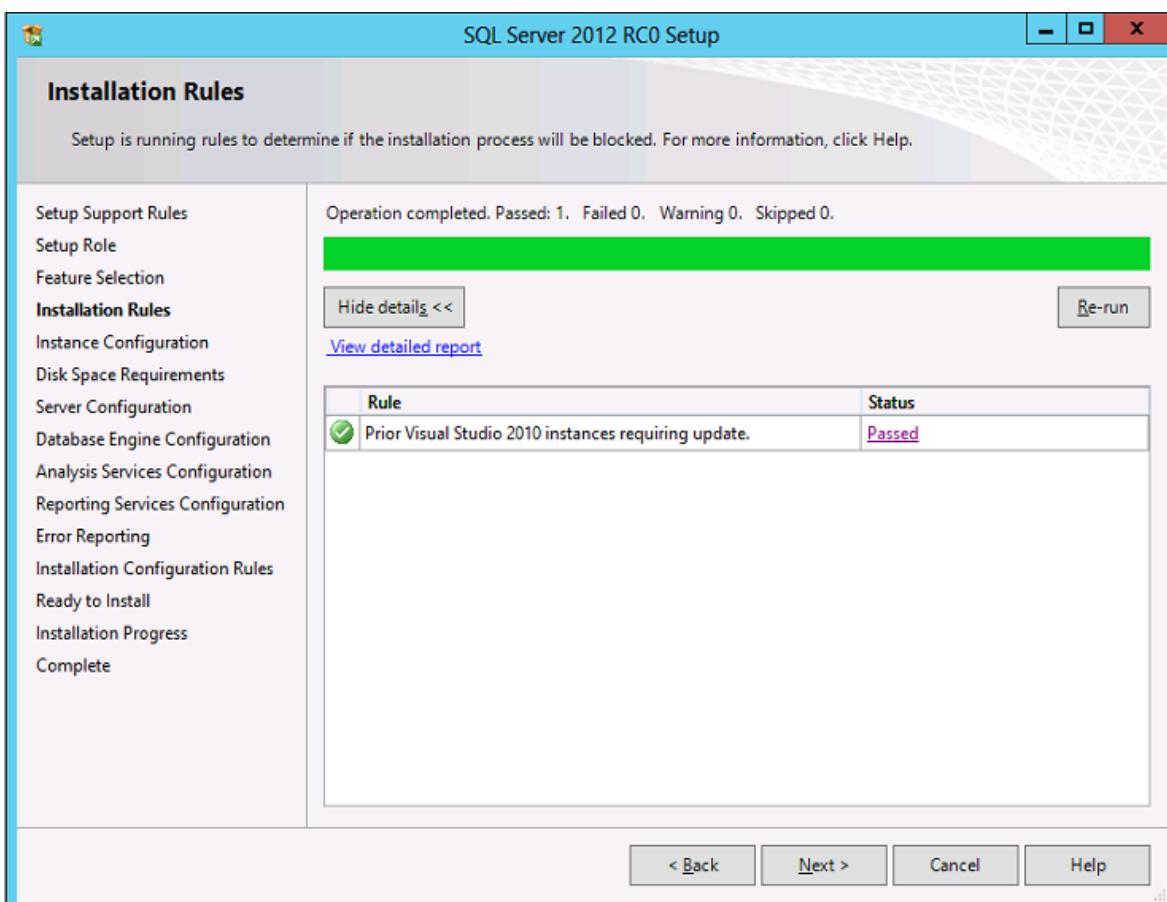
Tiếp tục chọn Next. Trong mục Features chọn các mục như hình dưới. Có thể lựa chọn hết tất cả.



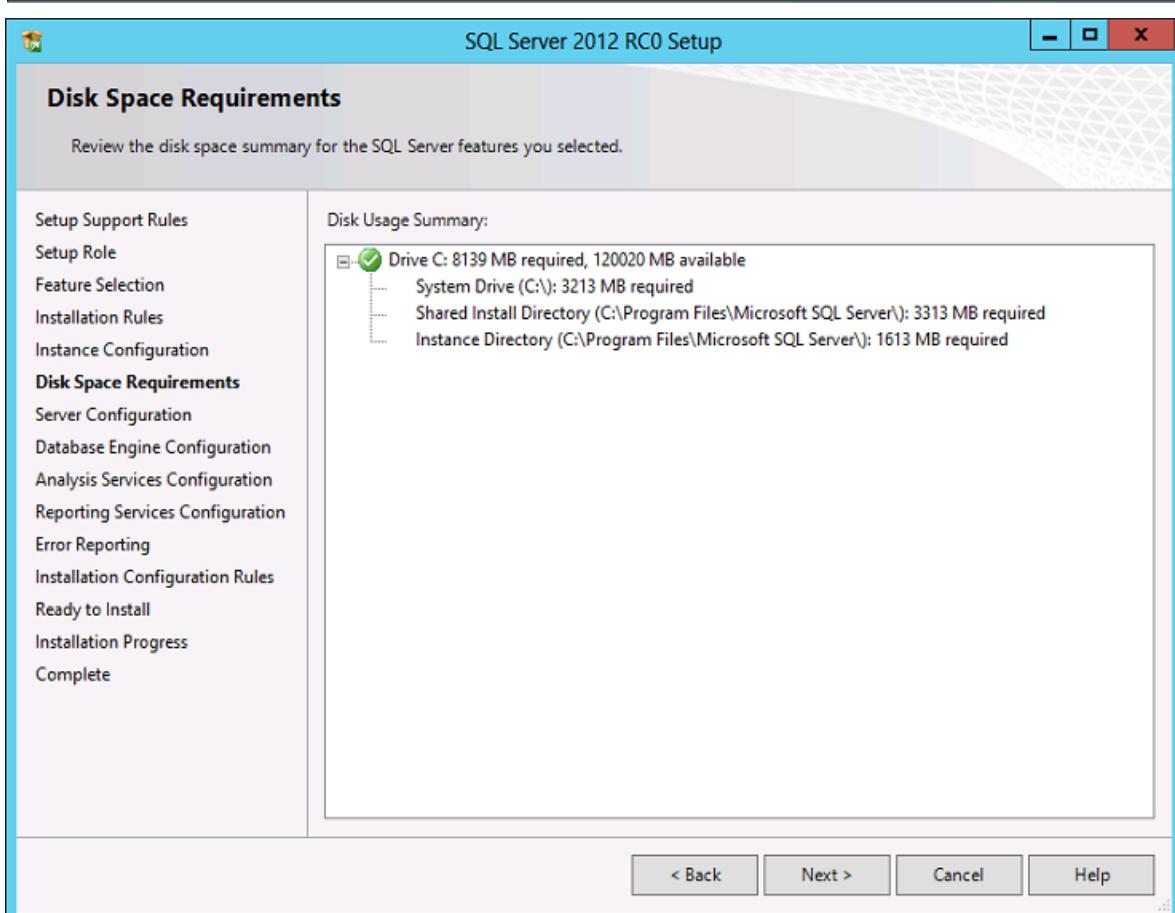
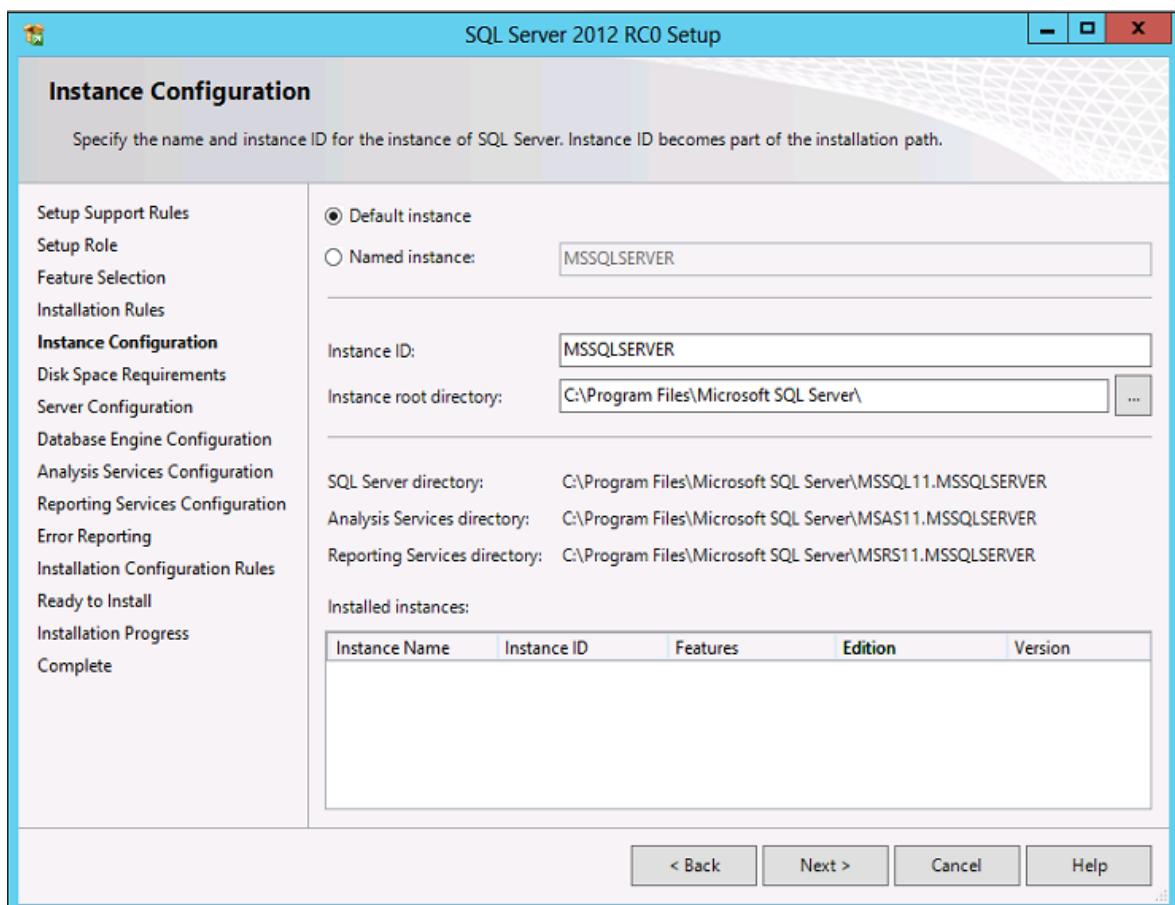
Tiếp tục chọn Next

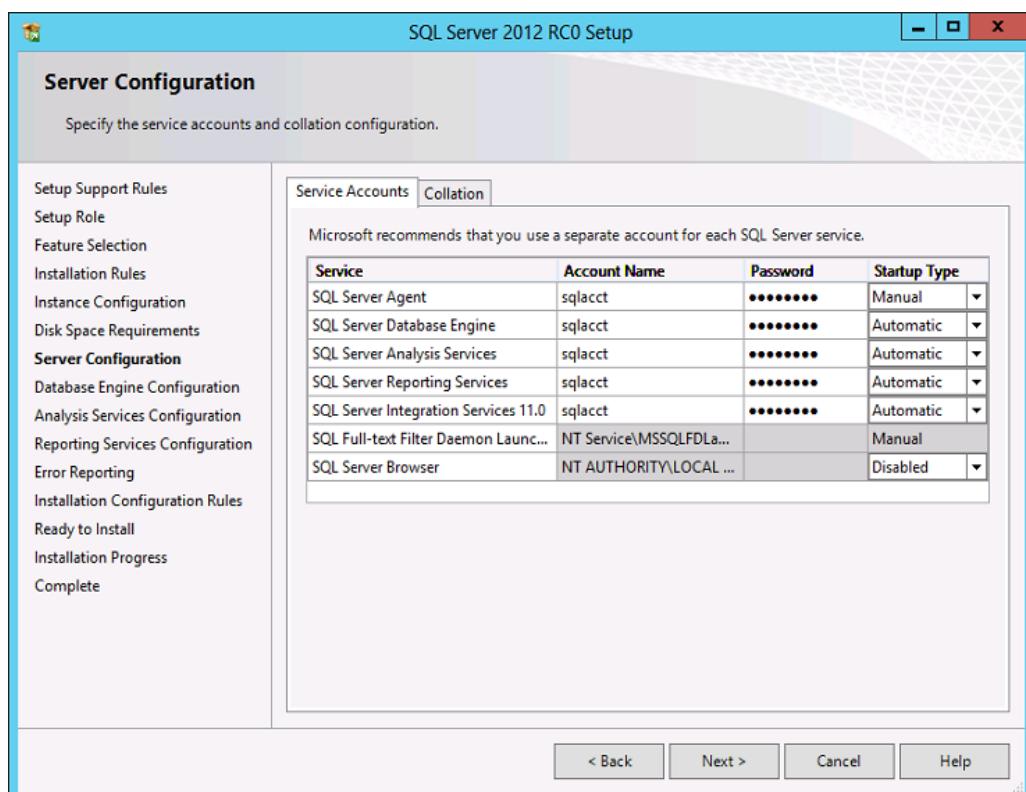


Tiếp tục chọn Next.



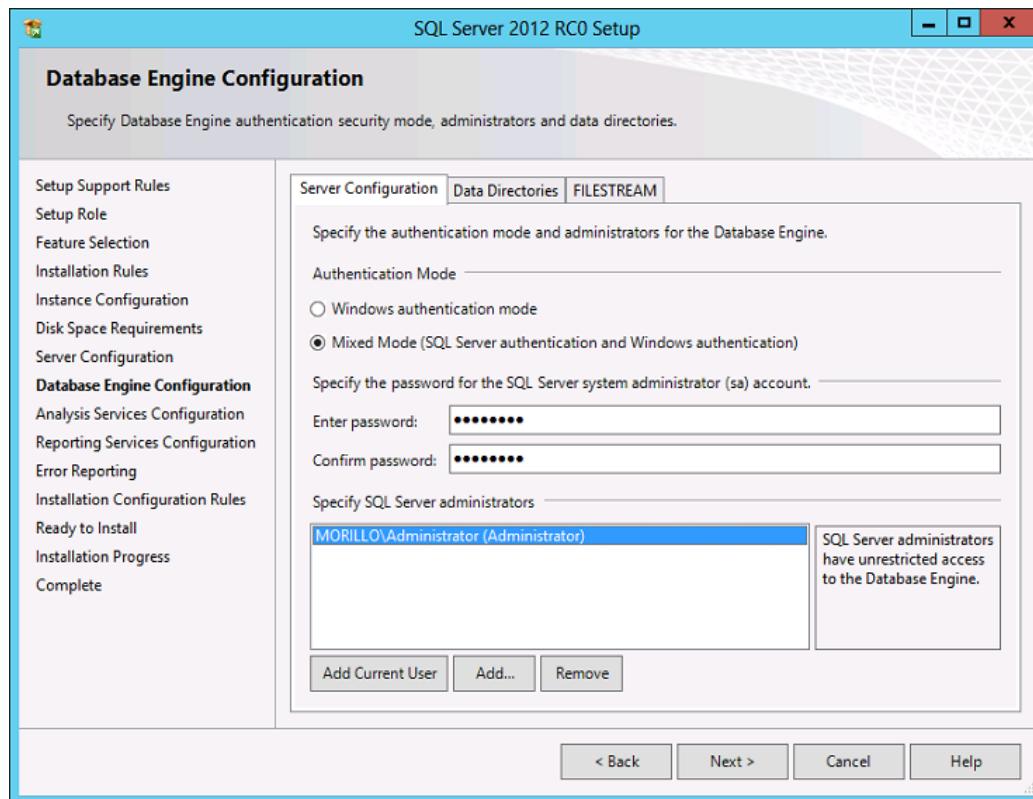
Tiếp tục chọn Next.





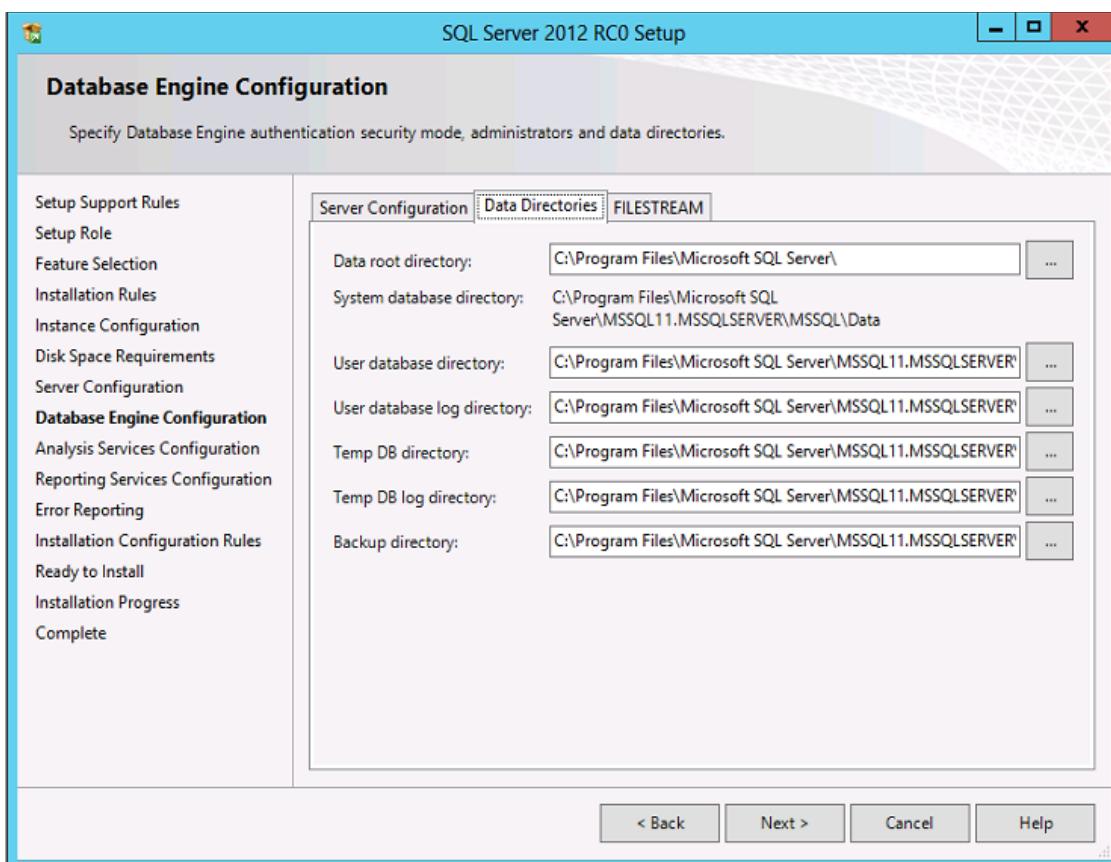
Tiếp theo màn hình sẽ xuất hiện ra hai lựa chọn:

- Windows Authentication mode: khi mở SQL Server để đăng nhập SQL Server sẽ tự động lấy Account Windows để đăng nhập.
- Mixed mode (SQL Server authentication and Windows authentication): phải tạo một password để đăng nhập SQL Server. Username mặc định là sa.

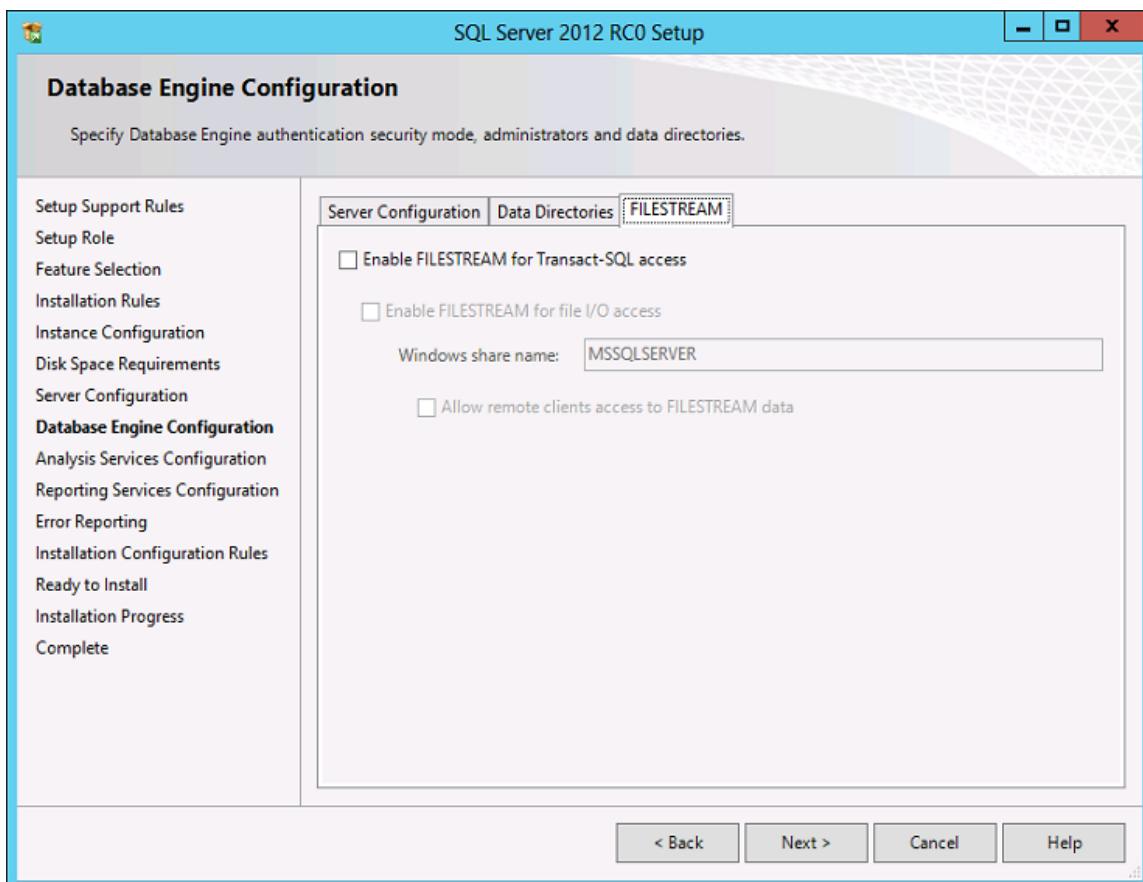


Hình 1.2. Giao diện Reporting Services Configuration

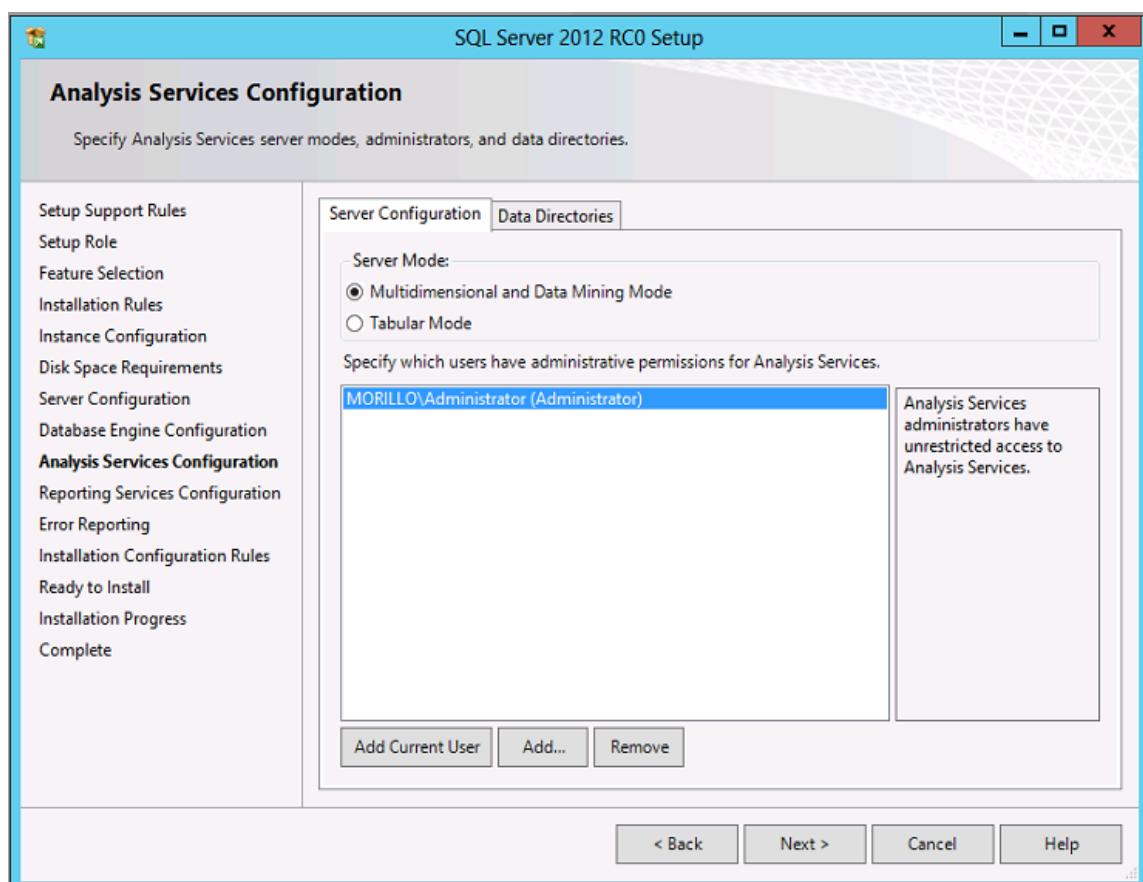
Tiếp tục chọn Next



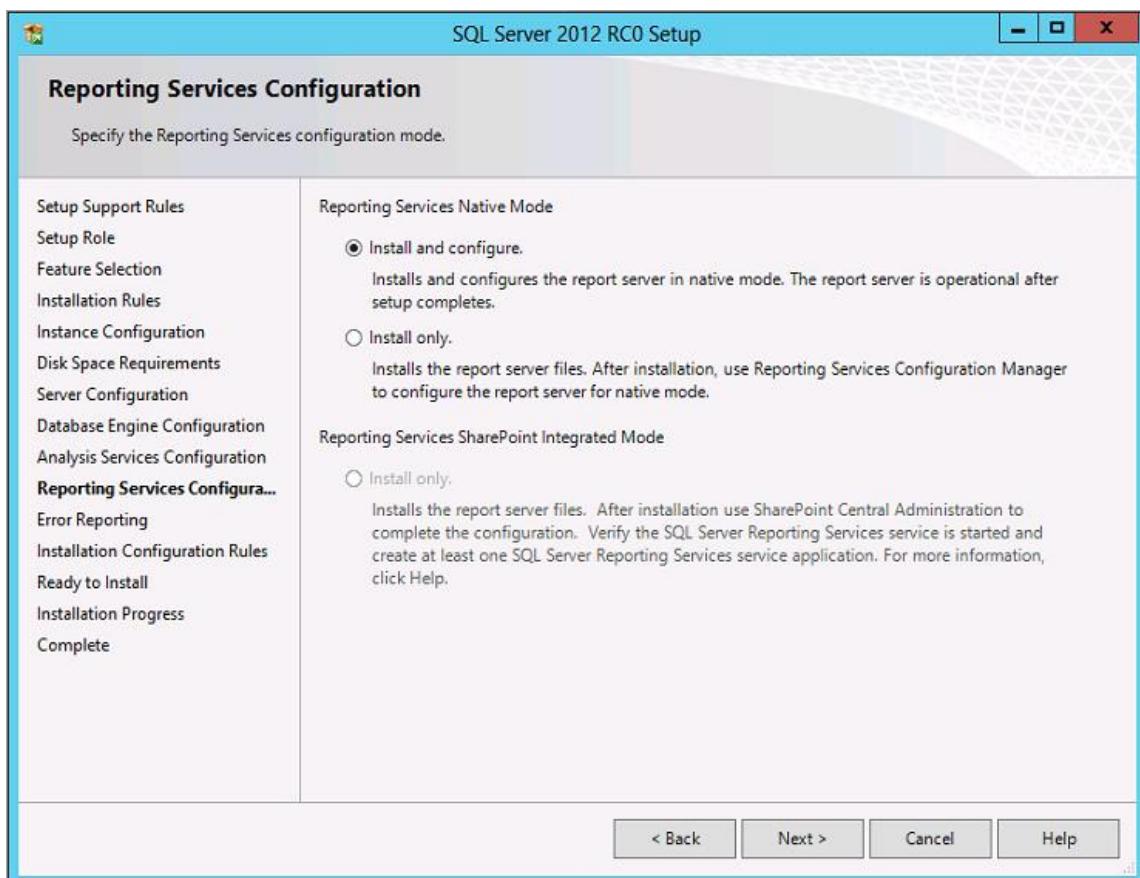
Chọn Next.



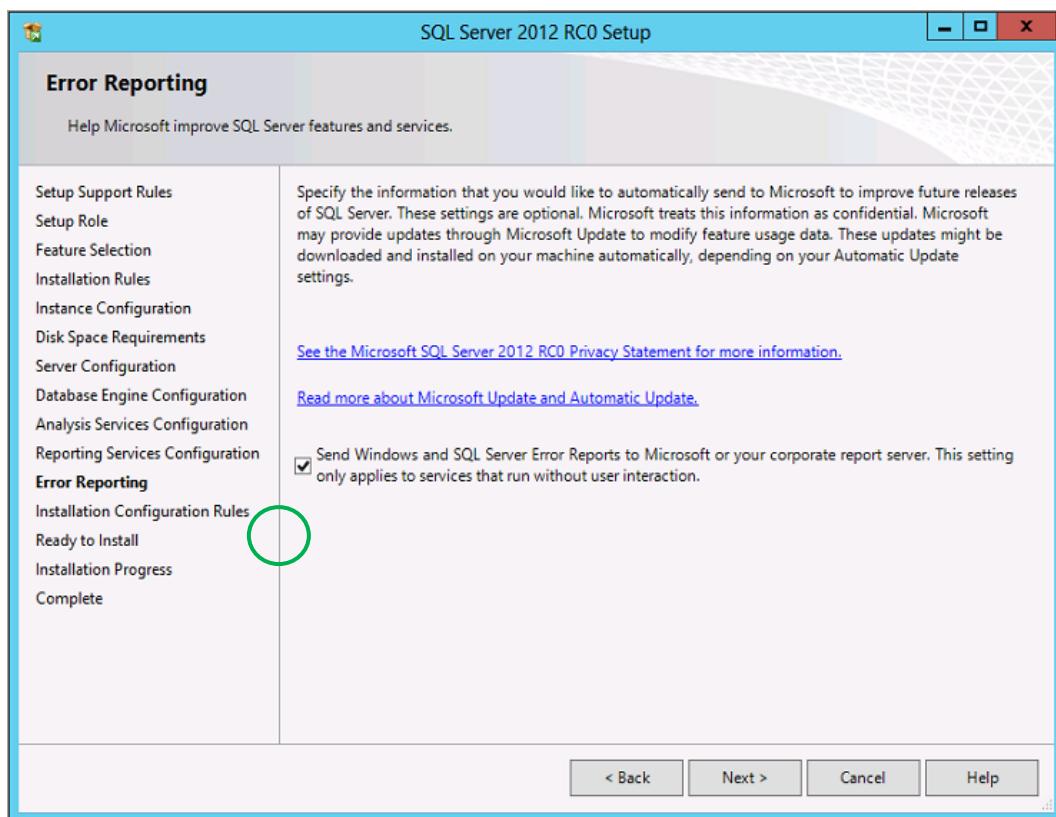
Chọn Next



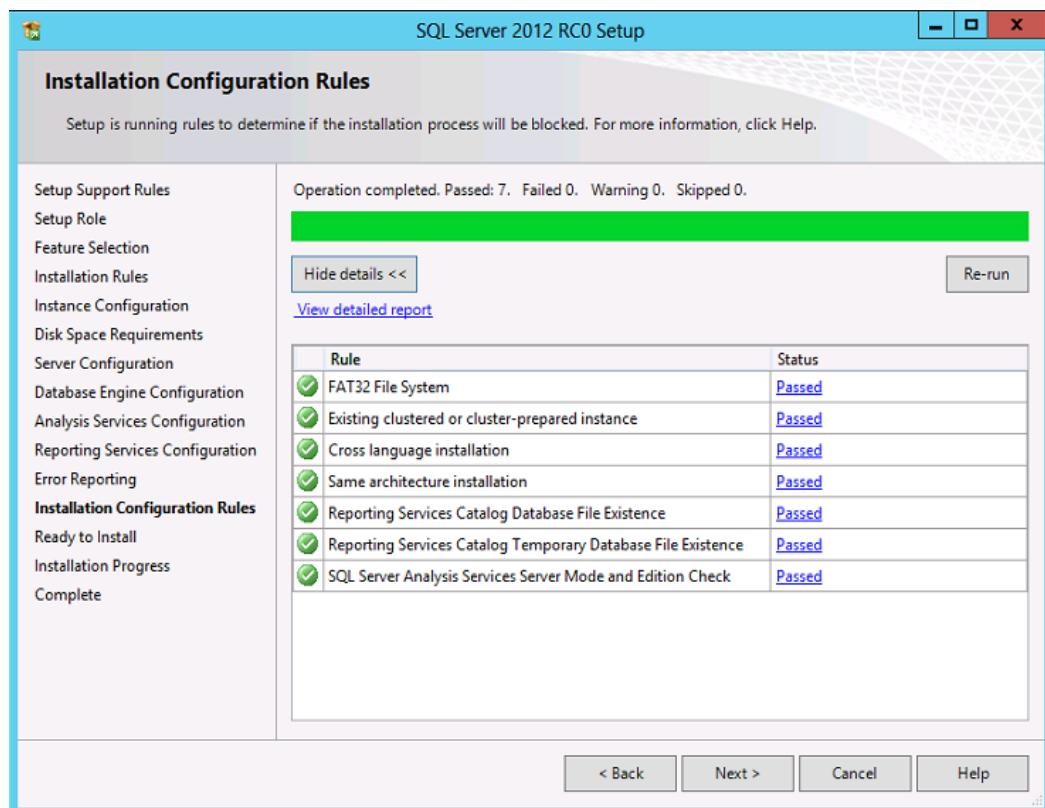
Chọn Next



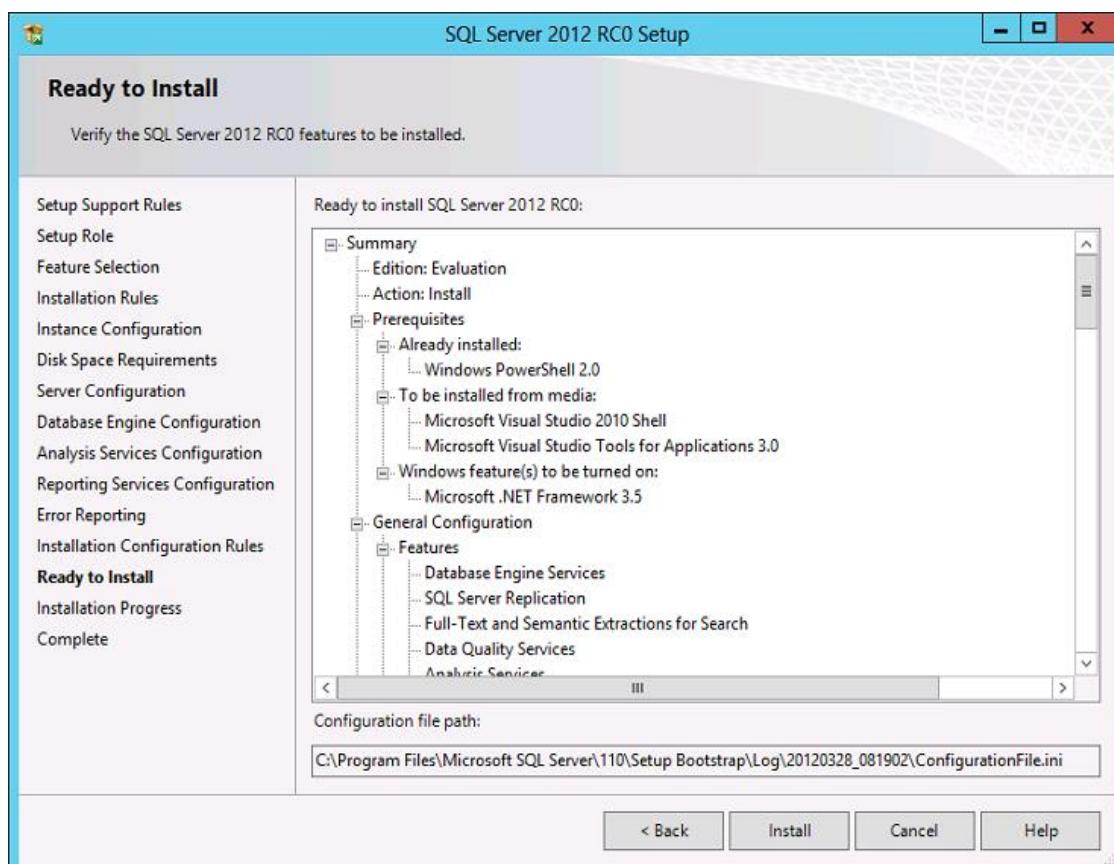
Tiếp theo, hình dưới đây bõ chọn mục *Send Windows and SQL Server Error Reports to Microsoft or your corporate report server. This setting only applies to services that run without user interaction.*



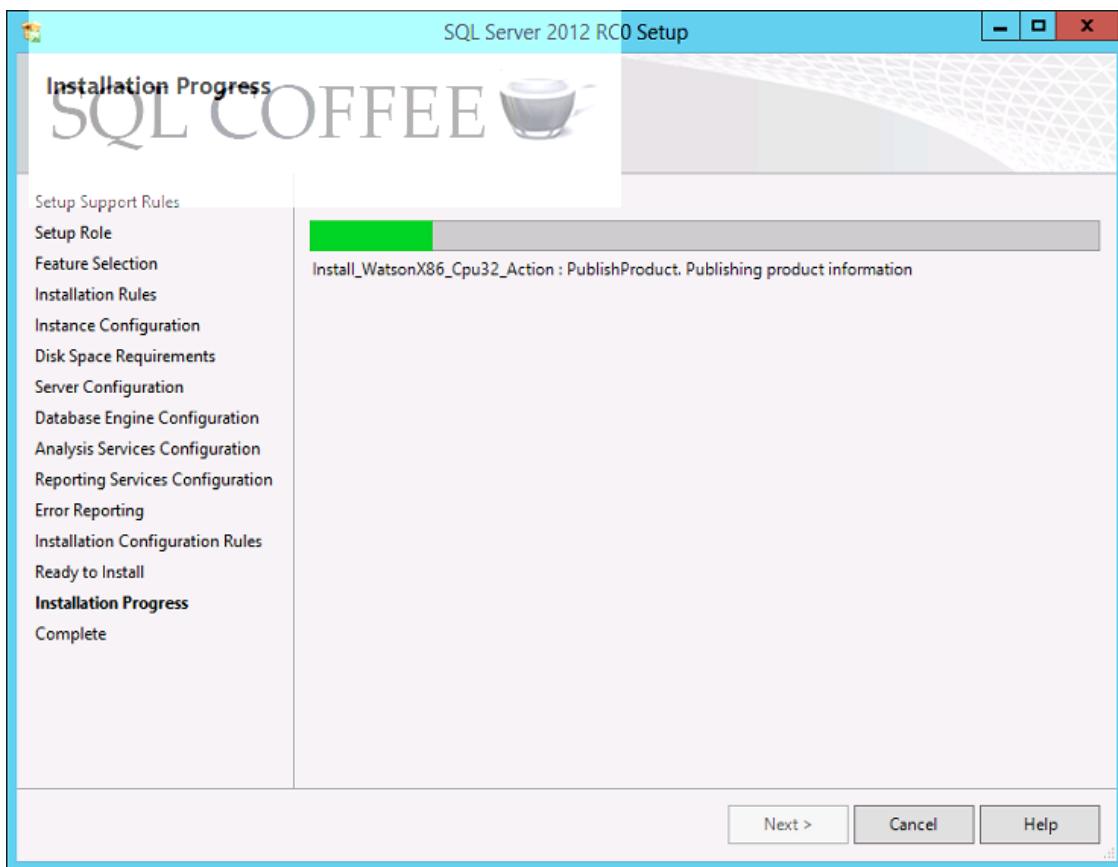
Chọn Next



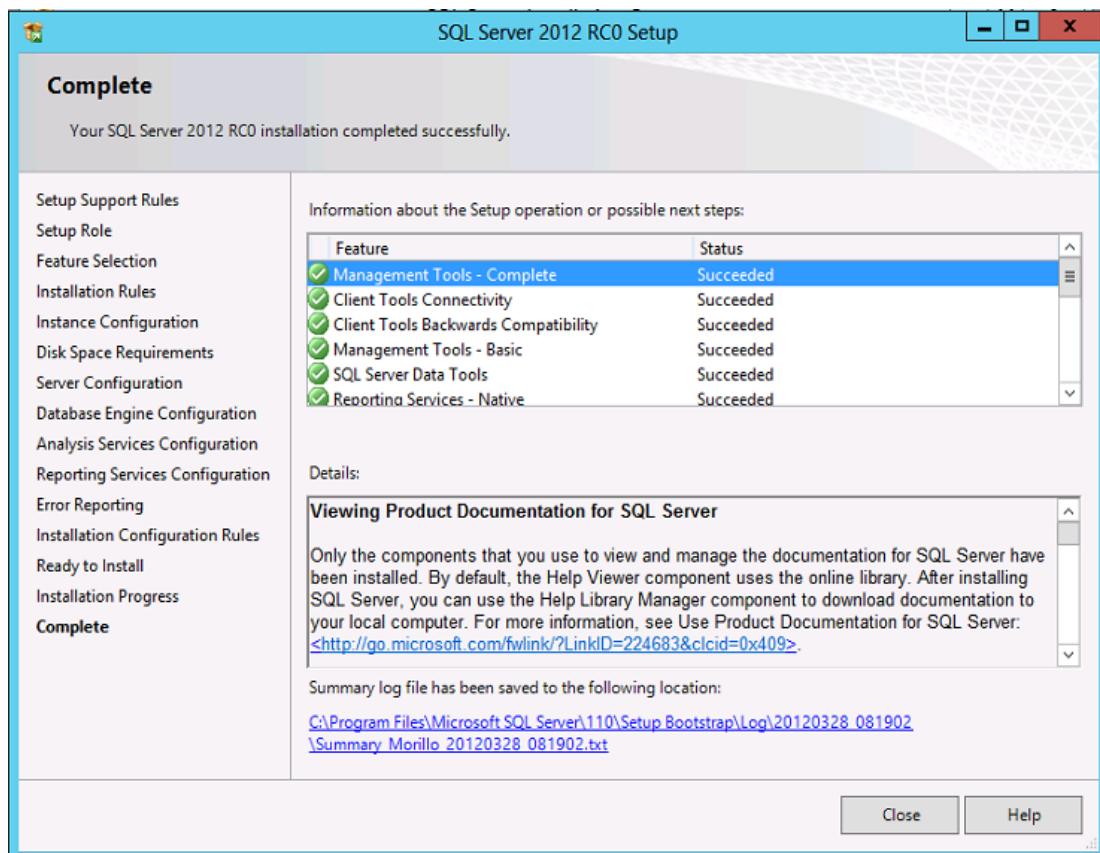
Chọn Install



Đến đây chờ khoảng 20 phút.



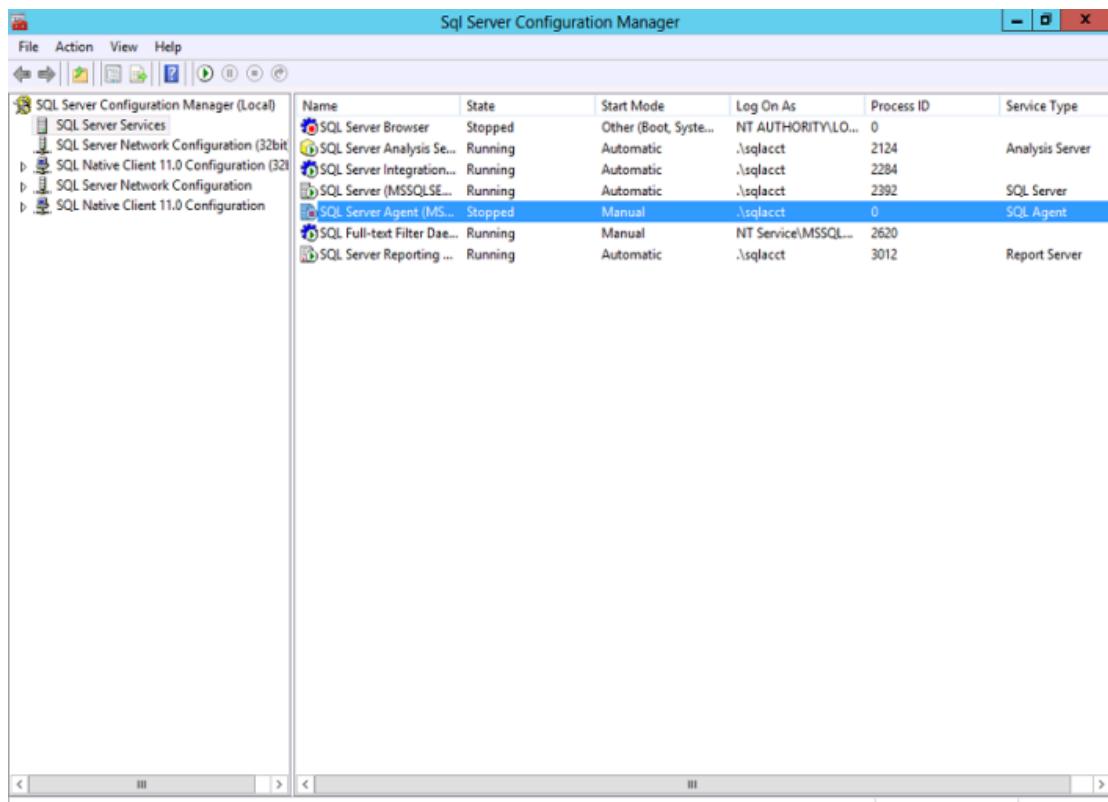
Kết quả như sau:



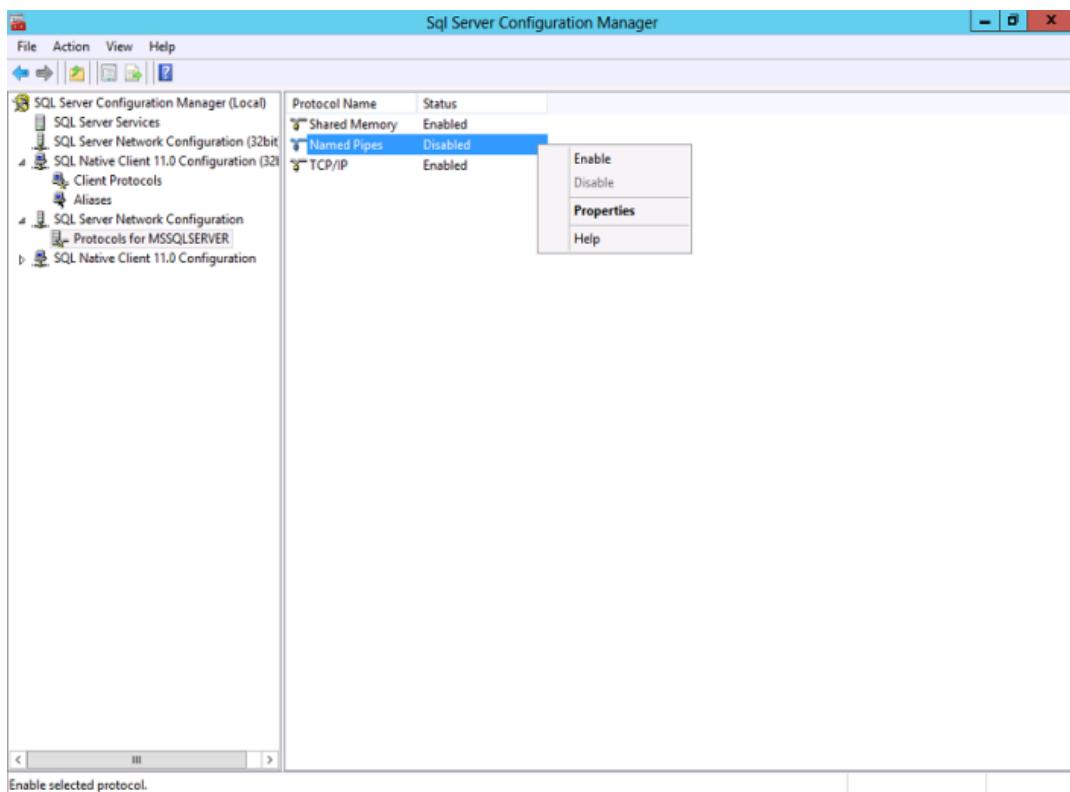
Chọn Close để kết thúc.

Sau khi cài đặt xong, thực hiện cấu hình cho SQL Server. Vào Control Panel → System and Security → Administrative Tools → Services.

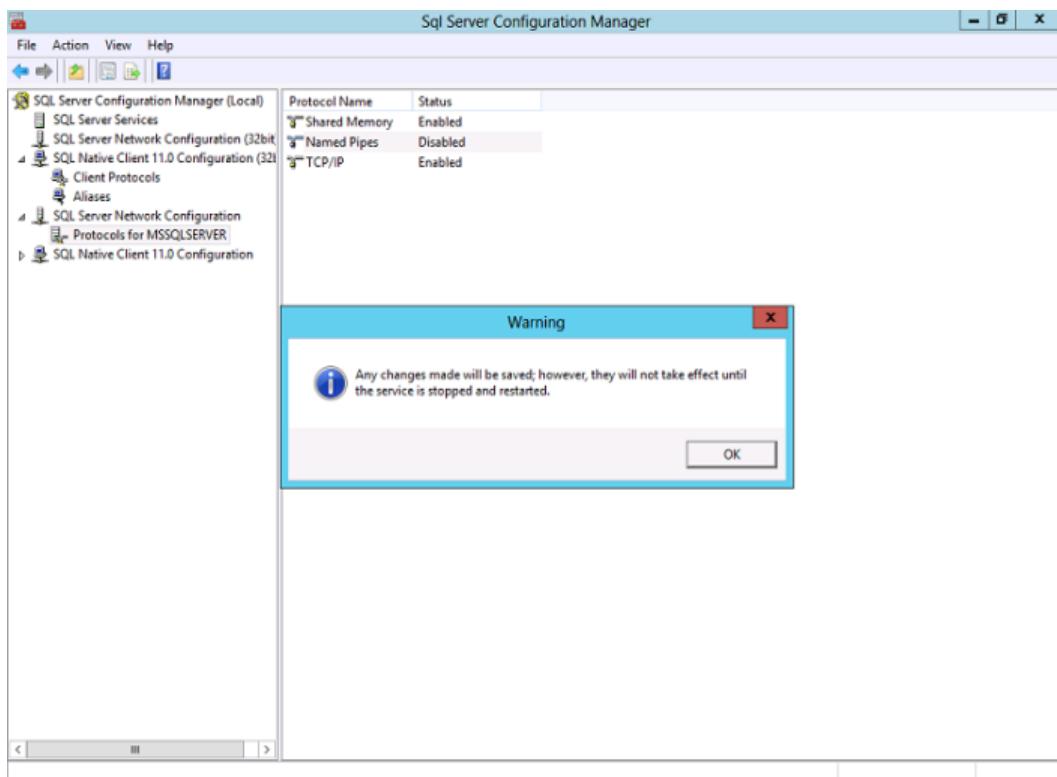
Chú ý: Nếu SQL Server Agent đang ở chế độ Stopped, cần phải chuyển sang chế độ Start.



Sau đó, tiếp tục thực hiện Enable cho các Protocol của SQL Server.



Sau khi thực hiện xong xuất hiện thông báo khởi động lại Service SQL Server (MSSQLSERVER), chọn OK.



Quá trình cài đặt đã hoàn thành!