NỘI DUNG THỰC TẬP

Nội dung thực tập 1: Xây dựng mô hình liên kết thực thể, mô hình quan hệ, cài đặt cơ sở dữ liệu, quản trị cơ sở dữ liệu

Mục đích: Giúp sinh viên phân tích dữ liệu và sử dụng hệ quản trị cơ sở dữ liệu SQL Server để cài đặt cơ sở dữ liệu.

Yêu cầu: Biết cách xây dựng mô hình liên kết thực thể, chuyển đổi sang mô hình quan hệ, tạo cơ sở dữ liệu, tạo bảng, nhập dữ liệu vào bảng bằng công cụ và bằng câu lệnh.

(Sinh viên đọc tài liệu tham khảo Giáo trình SQL Server từ trang 6-71, slide bài giảng Chương1_ Các khái niệm cơ bản (Nội dung mô hình liên kết thực thể, mô hình quan hệ), chương 4_ Hệ quản trị SQLServer và ngôn ngữ SQL (Nội dung: cách tạo cơ sở dữ liệu, tạo bảng)

I. Tóm tắt lý thuyết:

1. Xây dựng mô hình dữ liệu

Mô hình liên kết thực thể (ER):

– là một mô hình dữ liệu mức quan niệm, mô hình hóa dữ liệu và mối quan hệ giữa các đối tượng trong thế giới thực.

– là mô hình trung gian để chuyển những yêu cầu quản lý dữ liệu trong thế giới thực thành mô hình CSDL quan hệ

Thực Thể (entity)

Thực thể là một đối tượng tồn tại và phân biệt được.

Thuộc tính (attribute)

Các đặc điểm riêng của thực thể gọi là các thuộc tính.

Chẳng hạn các thuộc tính của sinh viên Nguyễn Văn Thành là:mã số sinh viên, giới tính, ngày sinh, hộ khẩu thường trú, lớp đang theo học, …

Loại thực thể(entity type)

Là tập hợp các thực thể có cùng thuộc tính. Mỗi loại thực thể đều phải được đặt tên sao cho có ý nghĩa. Một loại thực thể được biểu diễn bằng một hình chữ nhật.

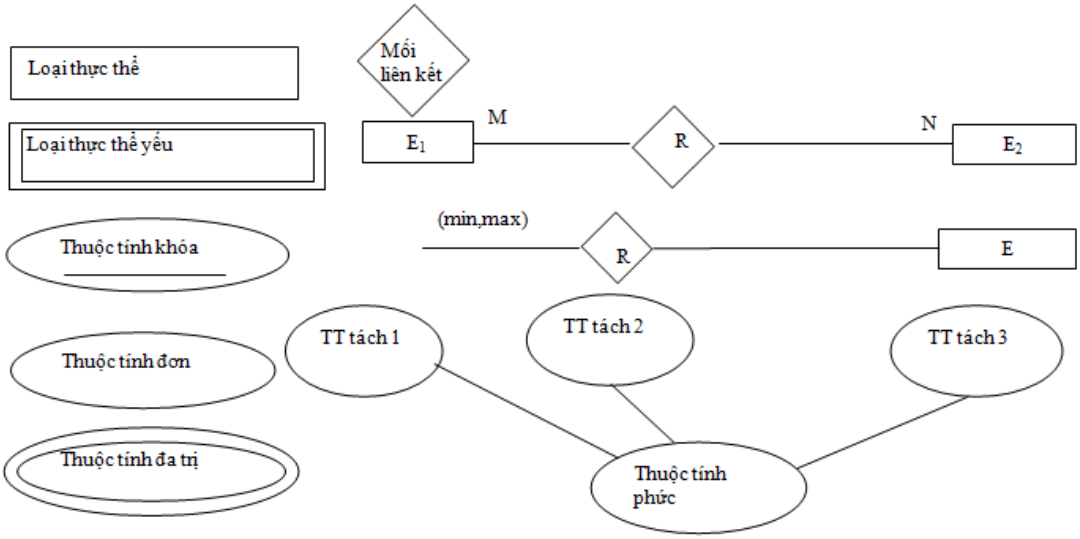
- ***Loại thuộc tính***
  - Thuộc tính đơn – không thể tách nhỏ ra được
  - Thuộc tính phức hợp – có thể tách ra thành các thành phần nhỏ hơn
- ***Loại giá trị của thuộc tính***

  Đơn trị: các thuộc tính có giá trị duy nhất cho một thực thể (VD: số CMND, …)

  Đa trị: các thuộc tính có một tập giá trị cho cùng một thực thể (VD: bằng cấp)

  Suy diễn: (năm sinh ← → tuổi)
- ***Quan hệ(liên kết):*** Là sự liên kết giữa 2 hay nhiều tập thực thể
- Ví dụ giữa tập thực thể NHANVIEN và PHONGBAN có các liên kết
  - Một nhân viên thuộc một phòng ban nào đó
  - Một phòng ban có một nhân viên làm trưởng phòng

  Các kí hiệu trong mô hình liên kết thực thể:



Mô hình quan hệ:

Mô hình quan hệ biểu thị cơ sở dữ liệu như một tập các quan hệ. Mỗi quan hệ có thể được biểu diễn như một bảng giá trị, mỗi một dòng trong bảng biểu thị một tấp hợp các giá trị dữ liệu liên quan với nhau.

- ***Quan hệ gồm***
  - Tên
  - Tập hợp các cột
    - Cố định
    - Được đặt tên
    - Có kiểu dữ liệu
  - Tập hợp các dòng
    - Thay đổi theo thời gian
  - Một dòng ~ Một thực thể
  - *Quan hệ ~ Tập thực thể*
- ***Thuộc tính:***
  - Tên các cột của quan hệ
  - Mô tả ý nghĩa cho các giá trị tại cột đó
  - Tất cả các dữ liệu trong cùng 1 một cột đều có cùng kiểu dữ liệu
- ***Lược đồ quan hệ***
  - Tên của quan hệ
  - Tên của tập thuộc tính
- ***Lược đồ CSDL***
  - Gồm nhiều lược đồ quan hệ
- ***Bộ:***
  - Là các dòng của quan hệ (trừ dòng tiêu đề - tên của các thuộc tính)
  - Thể hiện dữ liệu cụ thể của các thuộc tính trong quan hệ

Quy tắc chuyển đổi từ mô hình liên kết thực thể sang mô hình quan hệ

Tên thực thể chuyển thành tên quan hệ

Tên thuộc tính chuyển thành tên thuộc tính của quan hệ

Thuộc tính khóa chuyển thành khóa chính của quan hệ

Tách thuộc tính đa trị(nếu có) thành một bảng độc lập, xét lại mối quan hệ với bảng ban đầu.

- **Quy tắc 1: Với kiểu liên kết 1:1**
  - Cách 1: Chuyển khóa chính của LĐQH này sang làm khóa ngoại của LĐQH kia hoặc ngược lại.
  - Cách 2: Nhập 2 kiểu thực thể và mối liên kết thành 1 LĐQH, chọn khóa chính cho phù hợp.
- **Quy tắc 2**: Với kiểu liên kết 1:n

Chuyển khóa chính của LĐQH bên 1 sang làm khóa ngoại của LĐQH bên nhiều.

- **Quy tắc 3:** Với kiểu liên kết n:n

Chuyển mối liên kết thành một LĐQH có thuộc tính là thuộc tính của mối liên kết, thêm các thuộc tính khóa chính của các LĐQH có liên quan, khóa chính của LĐQH mới này là các thuộc tính mới thêm vào.

- **Quy tắc 4:** Thực thể yếu Chuyển thành một quan hệ: Có cùng tên với thực thể yếu. Thêm vào thuộc tính khóa của quan hệ liên quan

2. Tạo cơ sở dữ liệu bằng công cụ trên Hệ quản trị SQLServer:

Tạo cơ sở dữ liệu theo công cụ:

- Vào Enterprise Manager -> Databases.
- Nhấn nút phải chuột/hoặc menu Action -> New Database…

Tạo bảng dữ liệu:

Table (bảng dữ liệu) là một thành phần cơ bản của CSDL, một CSDL được thiết kế từ một hoặc nhiều bảng dữ liệu, mỗi bảng dữ liệu được cấu trúc từ các hàng và cột dữ liệu, mỗi hàng dùng mô tả một đối tượng, vấn đề, sự kiện,... cột thể hiện thuộc tính của các đối tượng, sự kiện,... của hàng. Dữ liệu cùng cột có cùng kiểu (data type). Ngoài các hàng, cột bảng còn có các khóa, liên kết, ràng buộc,...

Trước khi tạo bảng phải xác định xem bảng sẽ xây dựng như thế nào, dựa trên một số thông tin sau:

- Kiểu dữ liệu trong bảng.
- Các cột, kiểu dữ liệu tương ứng (và độ dài nếu cần thiết).
- Cột nào cho phép giá trị NULL (là giá trị mà phần dữ liệu thuộc hàng, cột xác định không được gán giá trị nào)
- Giá trị ngầm định (là giá trị mà khi chưa nhập vào nó nhận giá trị này).
- Chỉ số Index, khóa chính, khóa ngoài.

Kiểu dữ liệu

SQL Server gồm những kiểu dữ liệu chính sau:

- **Integers**
  - Bigint: 8 bytes
  - Int: 4 bytes
  - Smallint: 2 bytes
  - Tinyint: 1 byte, từ 0 -> 255.
- **bit**
  - Bit: 1 hoặc 0 value.

- **decimal and numeric**
  - Decimal từ -10^38+1->10^38 –1.
  - Numeric: giống **decimal.**
- **money and smallmoney**
  - Money: 8 bytes
  - Smallmoney: 4 bytes
- **Approximate Numerics**
  - Float: từ -1.79E + 308 -> 1.79E + 308.
  - Real: từ -3.40E + 38 -> 3.40E + 38.
- datetime and smalldatetime
  - Datetime: từ 1/1/1753-> 31/12/9999.
  - Smalldatetime từ 1/1/1900, -> 6/6/2079.
- Character Strings
  - Char: Fixed-length non-Unicode character, <= 8,000 ký tự
  - Varchar: Variable-length non-Unicode , <= 8,000 ký tự
  - Text: Variable-length non-Unicode <=2^31 - 1 (2,147,483,647) ký tự
- Unicode Character Strings
  - nchar Fixed-length Unicode , <=4,000 characters.
  - nvarcharVariable-length Unicode, <=4,000 characters
  - Ntext Variable-length Unicode <= 2^30 - 1 (1,073,741,823) characters.

Các khóa

Khóa chính – Primary Key: Là một hoặc tổ hợp nhiều cột dữ liệu xác định duy nhất trong một bảng, giá trị khóa chính luôn khác NULL.

Khóa ngoài(Foreign key): Cột dữ liệu là khóa ngoài có thể có quan hệ với nhiều khóa chỉnh ở nhiều bảng, một bảng có thể có nhiều khóa ngoài, khóa ngoài có thể có giá trị NULL, giá trị của khóa ngoài luôn nằm trong tập giá trị của khóa chính trong mối quan hệ đã thiết lập.(Khóa ngoài và khóa chính phải có cùng kiểu dữ liệu, cùng kích thước.)

Ràng buộc Unique

Unique là ràng buộc xác định trên một hoặc tổ hợp cột dữ liệu, cột hoặc tổ hợp cột dữ liệu được xác định ràng buộc loại này là duy nhất.

Một bảng dữ liệu có thể có nhiều ràng buộc duy nhất, một cột trongmrang buộc loại này cho phép nhận giá trị NULL, ràng buộc duy nhất có thể sử dụng làm tham chiếu cho khóa ngoài.

Ràng buộc Check

Là ràng buộc khống chế dữ liệu nằm trong một phạm vi nào đó. Ràng buộc này sẽ kiểm tra dữ liệu khi nhập vào.

Giá trị ngầm định – Default

Giá trị gán cho cột dữ liệu khi thêm bản ghi và chứa nhập dữ liệu vào cột này.

Tạo bảng

- Chọn CSDL
- Chọn Tables

- Nhấn phải chuột ở cửa sổ bên phải
- Chọn New Table.

Đặt khóa chính

Để xác định khóa chính ta thực hiện chọn những cột tham gia khóa bằng cách giữ phím shift và chọn chuột -> nhấn chuột phải -> chọn Set primary key.

Xác định Identity

- Chọn cột dữ liệu -> Chọn yes trong mục Identity -> đặt seed (giá trị khởi đầu) -> đặt increment (bước tự động tăng).

Sửa cấu trúc bảng

- Chọn bảng cần sửa đổi của CSDL.
- Nhất phải chuột -> chọn Design Table.
- Thực hiện sửa cấu trúc bảng.

Xóa bảng

- Chọn bảng
- Nhất chuột phải
- Chọn Delete -> Yes.

Bảng dữ liệu có tham gia mối quan hệ Relationship khi xóa bạn cần chú ý: Nếu bảng chứa khóa ngoài thì việc xóa thực hiện bình thường, nếu bảng chứa khóa chính của mối quan hệ thì không xóa được.

Nhập dữ liệu vào bảng

Sử dụng công cụ.

- Chọn bảng dữ liệu
- Nhất chuột phải -> Open Table

3. Tạo bảng bằng câu lệnh:

a. Lệnh định nghĩa dữ liệu DDL

- Là ngôn ngữ mô tả
  - Lược đồ cho mỗi quan hệ
  - Miền giá trị tương ứng của từng thuộc tính
  - Ràng buộc toàn vẹn
  - Chỉ mục trên mỗi quan hệ
- Gồm
  - CREATE TABLE (tạo bảng)
  - DROP TABLE (xóa bảng)
  - ALTER TABLE (sửa bảng)
  - CREATE DOMAIN / CREATE TYPE (tạo miền giá trị)
  - CREATE DATABASE
  - CREATE INDEX
  - ....

Tạo bảng:

- Để định nghĩa một bảng
  - Tên bảng
  - Các thuộc tính

- Tên thuộc tính
- Kiểu dữ liệu
- Các ràng buộc toàn vẹn trên thuộc tính (RBTV)

- Cú pháp: CREATE TABLE <Tên_bảng> (
  <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
  <Tên_cột> <Kiểu_dữ_liệu> [<RBTV>],
  .....
  [<RBTV>])
- **<RBTV>**
  - NOT NULL
  - NULL
  - DEFAULT: quy định giá trị mặc định trên các cột
  - UNIQUE, PRIMARY KEY, FOREIGN KEY / REFERENCES: tạo nên tính toàn vẹn thực thể một bảng dữ liệu và tính toàn vẹn tham chiếu giữa các bảng trong cơ sở dữ liệu
  - CHECK: quy định giá trị dữ liệu hay khuôn dạng dữ liệu được cho phép chấp nhận trên các cột của bảng.

Lệnh sửa bảng: được dùng để thay đổi cấu trúc bảng, thay đổi RBTV

Thêm cột:

ALTER TABLE <tên_bảng> ADD <Tên_cột> <Kiểu_DL> [<RBTV>]

Xóa cột:

ALTER TABLE <tên_bảng> DROP COLUMN <Tên_cột>

Mở rộng cột:

ALTER TABLE <tên_bảng> ALTER COLUMN <Tên_cột> <Kiểu_DL_mới> [<RBTV>]

Thêm RBTV:

ALTER TABLE <Tên_bảng> ADD CONTRAINT <Tên_RBTV> <RBTV>,
CONTRAINT <Tên_RBTV> <RBTV>, ...

Xóa RBTV

ALTER TABLE <Tên_bảng> DROP <Tên_RBTV> <Tên_RBTV>

Lệnh xóa bảng: Drop Table <Tên_bảng>

Ví dụ:

Tạo bảng PHONGBAN:

Cách 1:

```
CREATE TABLE PHONGBAN (
    MAPB NCHAR(10) PRIMARY KEY,
    TENPB VARCHAR(20) UNIQUE,
    MATP NCHAR(10),
    NG_NHANCHUC DATETIME DEFAULT (GETDATE())
)
```

Cách 2:

```
CREATE TABLE PHONGBAN (
    MAPB NCHAR(10) NOT NULL,
    TENPB VARCHAR(20) UNIQUE,
```

MATP NCHAR(10),
NG_NHANCHUC DATETIME DEFAULT (GETDATE())
PRIMARY KEY (MAPB)
)

Tạo bảng DUAN:

CREATE TABLE DUAN(
MADA NCHAR(10) PRIMARY KEY,
TENDA VARCHAR(20) NOT NULL,
MAPB NCHAR(10) REFERENCES PHONGBAN(MAPB)
)

Tạo bảng NHANVIEN:

CREATE TABLE NHANVIEN (
MANV NCHAR(10) PRIMARY KEY,
NS DATETIME,
DCHI VARCHAR(50),
GT CHAR(3) CHECK (GT IN ('Nam', 'Nu')),
LUONG INT DEFAULT (10000),
MA_NGS NCHAR(10),
MAPB NCHAR(10) REFERENCES PHONGBAN(MAPB)
)

Tạo bảng PHANCONG:

CREATE TABLE PHANCONG (
MANV NCHAR(10) REFERENCES NHANVIEN(MANV),
MADA NCHAR(10) REFERENCES DUAN(MADA),
SOGIO DECIMAL(3,1),
CONSTRAINT NV_DA PRIMARY KEY(MANV,MADA)
)

Tạo bảng THANHNHAN:

```
create table thannhan(
manv char(10) references nhanvien(manv),
hoten nvarchar(50),
ngaysinh date,
gioitinh nchar(6) check(gioitinh in (N'Nam',N'Nữ')),
primary key(manv,hoten)
)
Ví dụ về constraint
create table phancong
 (manv char(10))

 alter table phancong
 add mada char(10) not null,
     sogio int

alter table phancong
alter column manv char(10) not null
```

```
alter table phancong
add primary key(manv,mada)


alter table phancong
add constraint fk_nv
foreign key (Manv)
references Nhanvien(manv)
```

b. Lệnh thao tác dữ liệu DML: Lệnh cập nhật dữ liệu: Insert, Update, Delete

– **Thêm dữ liệu:**

Thêm 1 dòng:

INSERT INTO <tên bảng>(<danh sách các thuộc tính>)

VALUES (<danh sách các giá trị>)

Thêm nhiều dòng:

INSERT INTO <tên bảng>(<danh sách các thuộc tính>)

<câu truy vấn con>

– **Xóa dữ liệu:**

DELETE FROM <tên bảng>

[**WHERE** <điều kiện>]

– **Sửa dữ liệu:**

UPDATE <tên bảng>

SET <tên thuộc tính>=<giá trị mới>,

<tên thuộc tính>=<giá trị mới>,

…

[**WHERE** <điều kiện>]

4. Quản trị cơ sở dữ liệu

Xác thực người dùng

- Các kiểu xác thực:
  - SQL Server and Windows Authentication: hỗ trợ 2 kiểu đăng nhập trên SQL Server và trên Windows
  - Windows Authentication
- Khi cài đặt chúng ta đã chọn một kiểu xác thực cho SQL Server. Tuy nhiên chúng ta có thể thay đổi:
  - Mở MSt.
  - Trong cửa sổ Object Explorer, ấn phải chuột lên server, chọn properties.
  - Chọn nút Security => chọn kiểu xác thực

- Chọn OK

- Kiểu Windows Authentication
  - Chỉ yêu cầu NSD đăng nhập một lần
  - Quản lý tập trung
  - Tận dụng được các tính năng bảo mật của Windows
- Kiểu SQL and Windows
  - Hỗ trợ NSD trên các HĐH khác
  - Quản lý NSD riêng
- Chú ý: Nếu sử dụng password policy thì mật khẩu phải tối thiểu là 6 kí tự và phải chứa cả ba loại: chữ thường a-z, chữ hoa A-Z, chữ số 0-9.
- Thêm NSD theo xác thực Windows
  - Mở Security mức Server
  - Ấn phải chuột chọn New Login
  - Lựa chọn: Windows Authentication
  - Nhập tên NSD, hoặc chọn Search
  - Chọn Default DB
  - Chọn Server Roles
  - Chọn User Mapping để chỉ ra các CSDL mà NSD có quyền tương tác.
- Thêm NSD theo xác thực SQL Server (làm tương tự)

Sao lưu dữ liệu

Có các cách sao lưu dữ liệu

- Detach/ Attach
- Backup/Restore
- Export/Import
- Generate Script

a. Detach/ Attach

- Detach:
  - Chọn database: chọn Tasks | Detach.

- Việc Detach CSDL có thể chưa sẵn sàng nếu như có ít nhất một user khác đang kết nối đến CSDL này.

- Attach:

  - Copy file dữ liệu và file log vào thư mục trong máy chủ. Thông thường đặt trong thư mục data trong sqlserver.

  - Trong cửa sổ Object Browser của Database chọn Attach.

  - Chương trình mở cửa số Attach. Chọn nút Add và mở thư mục chứa file dữ liệu (.mdf).

b. Backup/Restore

- Click chuột phải vào cơ sở dữ liệu cần Backup, chọn Task/Backup

- Click chuột phải vào cơ sở dữ liệu cần Restore, chọn Tasks/Restore/Database

- Có thể chọn Restore từ Database hoặc từ File.bak trong ổ đĩa

c. Generate Scipts

- Click chuột phải vào cơ sở dữ liệu cần sinh scipts, chọn Task/Generate Scipts..

- Chọn Next, đến mục Specify how scipts should be saved or published, chọn Advanced. Tại dòng Types of data to scipt Chọn Schema and data (Cả cấu trúc tạo bảng và lệnh insert dữ liệu)

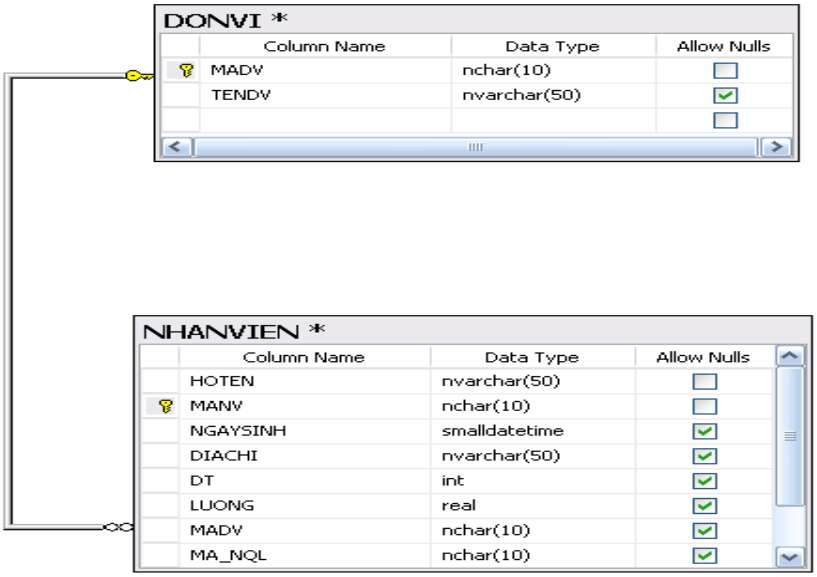
d. Export/Import: sinh viên tự tìm hiểu, chủ yếu export/ import đối với file excel

II. Bài thực hành:

Bài 1:

Sinh viên sử dụng công cụ trong SQL Server để thực hiện:

1. Tạo cơ sở dữ liệu QUANLYNHANSU.
2. Tạo 2 bảng Nhanvien và Donvi với cấu trúc các trường như trong hình dưới. Nhập dữ liệu cho 2 bảng này: ít nhất 5 bản ghi cho bảng Donvi. 10 bản ghi cho bảng nhân viên.



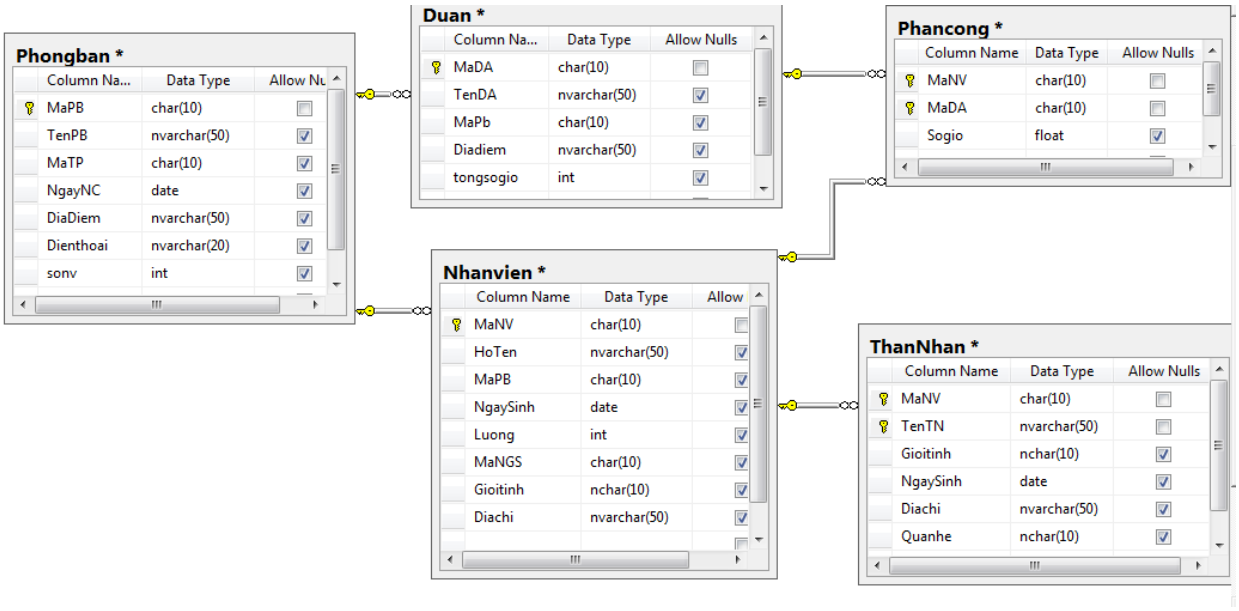
Bảng NHANVIEN

MANV	HOTEN	NGAYSINH	DIACHI	DIENTHOAI	HSLUONG	MADONVI
NV01001	Nguyễn Thị Hoa	05/05/1976	56 Lê Duẩn	521304	2.11	01
NV02001	Lê Hoài Nam	03/05/1976	32 Trần Phú	823145	1.86	02
NV02002	Hoàng Nam Phong	05/08/1971	66 Hoàng Diệu	521247	3.21	02
NV03001	Trần Nguyên Phong	20/12/1976	7 Hà Nội	849290	1.92	03
NV03002	Nguyễn Hữu Tình	18/08/1976	7 Hà Nội	849290	1.92	03
NV05001	Nguyễn Trung Kiên	14/05/1972	77 Nguyễn Huệ	823236	1.86	05

Bảng DONVI

MADONVI	TENDONVI	DIENTHOAI
01	Phòng Kế toán	821451
02	Phòng Tổ chức	831414
03	Phòng điều hành	823351
04	Phòng đối ngoại	841457
05	Phong Tài vụ	821451

Bài 2:
Sinh viên tạo cấu trúc các bảng như sơ đồ bên dưới. Nhập dữ liệu cho các bảng.



Bài 3: Sinh viên làm theo nhóm thực hiện đề tài tự chọn (mỗi nhóm 3-4 sinh viên). Cụ thể: mô tả bài toán, xây dựng mô hình liên kết thực thể, chuyển đổi sang mô hình quan hệ, cài đặt cấu trúc dữ liệu và nhập dữ liệu (thao tác bằng công cụ).

Danh sách các đề tài tự chọn:

1. Quản lý nhà hàng
2. Quản lý sinh viên
3. Quản lý thư viện
4. Quản lý khu vui chơi giải trí

Bài 4: Sinh viên thực hành câu lệnh cập nhật dữ liệu cho các bảng trong cơ sở dữ liệu QUANLYNHANVIEN. Thực hành các câu lệnh thêm, sửa, xóa dữ liệu.

Bài 5: Làm việc theo nhóm. Trên đề tài nhóm đã chọn. Thực hành câu lệnh để tạo các bảng trong cơ sở dữ liệu của nhóm. Dùng câu lệnh cập nhật dữ liệu cho các bảng. Thực hành các câu lệnh thêm, sửa, xóa dữ liệu.

Bài 6: Tạo người dùng và cấp quyền cho người dùng trên cơ sở dữ liệu Quản lý nhân viên. Thực hiện các loại sao lưu dữ liệu.

Bài 7: Thực hành tạo người dùng, cấp quyền, sao lưu dữ liệu trên đề tài nhóm.

Nội dung thực tập 2: Truy vấn và cập nhật dữ liệu

Mục đích: Giúp sinh viên làm quen và sử dụng được hệ quản trị cơ sở dữ liệu SQL Server. Thành thạo các câu lệnh truy vấn dữ liệu.

Yêu cầu: Sinh viên thành thạo các câu lệnh truy vấn dữ liệu. Nâng cao khả năng làm việc theo nhóm: viết các câu lệnh truy vấn dữ liệu cho các yêu cầu phù hợp với thực tế.

(Sinh viên đọc tài liệu tham khảo Giáo trình thực hành SQL từ trang 12-31, Slides bài giảng của Giáo viên: Chương 4. Hệ quản trị SQLServer và Ngôn ngữ SQL)

I. Tóm tắt lý thuyết:

Cú pháp:

Select <danh sách trường>

From <danh sách bảng>

Where <đk nối> and <điều kiện lọc>

Hoặc:

Select <danh sách trường>

From <Bảng 1> <[inner/left/right] join> <bảng 2> on <điều kiện nối>

Where <điều kiện lọc>

Ví dụ:

```
select manv, hoten, tenpb
from nhanvien ,phongban
where nhanvien.mapb=phongban.mapb and luong>5000000
```

```
select manv, hoten, tenpb
from nhanvien  join phongban  on  nhanvien.mapb=phongban.mapb where
luong>5000000
```

- Truy vấn trên một bảng:

```
Select manv,hoten
From nhanvien
```

- Truy vấn với Bí danh cho trường:

```
Select manv,hoten as 'Họ và tên'
    From nhanvien
```

- Biểu thức tính toán trên mệnh đề Select:

```
Select manv,hoten as 'Họ và tên', luong*1.1 luongtang
    From nhanvien
    Where mapb='PB01'
```

- Truy vấn trên nhiều bảng:

```
select manv,hoten
from nhanvien,phongban
where manv=matp and hoten like '%h%'
```

- Bí danh cho bảng:

```
select manv, hoten, tenpb
from nhanvien n,phongban p
where n.mapb=p.mapb
```

- Truy vấn lồng (Truy vấn con)

Ví dụ: Đưa ra mã nhân viên, họ tên của những nhân viên thuộc phòng ban có địa điểm ở TP HCM

```
SELECT manv, hoten
FROM NhanVien
WHERE MaPB IN
  (SELECT MaPB
    FROM PhongBan
    WHERE Diadiem='TP HCM' )
```

Ví dụ: Đưa ra mã, họ tên của những nhân viên thuộc phòng Nghiên cứu

```
SELECT manv,hoten
FROM Nhanvien N
WHERE EXISTS (
        SELECT *
        FROM Phongban
        WHERE MaPB=N.MaPB and TenPB='Nghien cuu')
```

Ví dụ: Đưa ra mã nhân viên, họ tên và tên phòng ban tương ứng của nhân viên

```
select manv,hoten,tenpb
from nhanvien, (select mapb,tenpb from phongban) as PB
where nhanvien.mapb=pb.mapb
```

Ví dụ: Tìm những nhân viên có lương bằng lương của ít nhất một nhân viên thuộc phòng 'PB01'

```
Select manv,hoten, luong
from nhanvien
where luong=any(select luong
          from nhanvien
          where mapb='PB01')
```

Ví dụ: Tìm tổng lương, lương cao nhất, lương thấp nhất và lương trung bình của các nhân viên

```
select tongluong=sum(luong), max(luong) as luong_max, min(luong) luong_min ,        avg(luong) luongTB
from nhanvien
```

Ví dụ: Đưa ra phòng ban có từ 20 nhân viên trở lên:

```
select mapb,count(manv)
from nhanvien
group by mapb
having count(manv)>=20
```

Ví dụ: Đưa ra phòng ban có đông nhân viên nhất:

```
Select mapb, count(manv)
from nhanvien
group by mapb
having count(manv)=
            (select max(sonv)
              From (
                  Select mapb, count(manv) sonv
                  from nhanvien
                  group by mapb
                  ) b)
```

Cách 2: Sử dụng View:

```
Create View V_DemNVTheoPB
as
Select mapb, count(manv) sonv
from nhanvien
group by mapb

Select mapb, count(manv)
from nhanvien
```

group by mapb

having count(manv)=

(select max(sonv)

From V_DemNVTheoPB)

Ví dụ: Truy vấn với inner join/ left join/ right join/ full join

Select <danh sách trường>

From <Bảng 1> <[inner/left/right] join> <bảng 2> on <điều kiện nối>

Where <điều kiện lọc>

Đưa ra danh sách nhân viên và tên phòng ban tương tứng của nhân viên

Select manv, hoten, tenpb

From Nhanvien inner join PhongBan on Nhanvien.MaPb=Phongban.MaPb

Đưa ra danh sách nhân viên và tên phòng ban tương tứng của nhân viên, đưa ra cả những nhân viên chưa thuộc phòng ban nào

Select manv, hoten, tenpb

From Nhanvien left join PhongBan on Nhanvien.MaPb=Phongban.MaPb

Đưa ra phòng ban và tên nhân viên tương ứng, đưa ra cả những phòng ban chưa có nhân viên nào.

Select manv, hoten, tenpb

From Nhanvien right join PhongBan on Nhanvien.MaPb=Phongban.MaPb

Đưa ra danh sách nhân viên và tên phòng ban tương ứng, đưa ra cả những nhân viên chưa thuộc phòng ban nào và những phòng ban chưa có nhân viên nào

Select manv, hoten, tenpb

From Nhanvien full join PhongBan on Nhanvien.MaPb=Phongban.MaPb

II. Bài thực hành:

Bài 1: Sinh viên thao tác với cơ sở dữ liệu quản lý đề án công ty, thực hiện các yêu cầu sau:

1. Đưa ra danh sách nhân viên (Mã nhân viên, họ tên, ngày sinh, địa chỉ, lương). Sắp xếp theo thứ tự lương giảm dần.
2. Đưa ra danh sách các phòng ban đã có trưởng phòng.
3. Đưa ra các phòng ban chưa có trưởng phòng.
4. Đưa ra danh sách các đề án đã có phòng ban chủ trì.
5. Đưa ra danh sách đề án chưa có phòng ban nào chủ trì.
6. Đưa ra danh sách nhân viên có người quản lý (người giám sát)
7. Đưa ra danh sách nhân viên không có người quản lý.
8. Đưa ra danh sách nhân viên (mã nhân viên, tên, ngày sinh, lương, tên phòng), sắp xếp theo tên phòng ban.
9. Đưa ra danh sách nhân viên thuộc phòng 'Hành chính'
10. Đưa ra danh sách nhân viên có thân nhân.
11. Đưa ra danh sách nhân viên có lương từ 5 triệu trở lên. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương, tên phòng ban.
12. Đưa ra đanh sách nhân viên có tham gia đề án. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương.

13. Đưa ra đanh sách nhân viên có tham gia đề án. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương, tên đề án, số giờ.
14. Đưa ra đanh sách nhân viên không tham gia đề án nào. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương.
15. Đưa ra danh sách đề án do phòng 'Hành chính' quản lý.
16. Đưa ra danh sách đề án mà nhân viên 'Nguyễn Thị Nga' thuộc phòng 'Hành chính' tham gia.
17. Đưa ra danh sách nhân viên có thân nhân cùng tên, cùng giới tính.
18. Đưa ra danh sách nhân viên có cùng họ tên, cùng giới tính.
19. Đưa ra danh sách các nhân viên là trưởng phòng.
20. Đưa ra danh sách các nhân viên là người quản lý.

Bài 2: Thực hành theo đề tài nhóm. Thực hiện truy vấn dữ liệu trên 1 bảng, 2 bảng, 3 bảng,...

Bài 3: Sinh viên thao tác với cơ sở dữ liệu quản lý đề án công ty, thực hiện các yêu cầu sau:

1. Đưa ra danh sách nhân viên có lương lớn nhất trong công ty.(Mã nhân viên, họ tên, ngày sinh, địa chỉ, lương).
2. Đưa ra tổng lương theo từng phòng ban. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, tổng lương.
3. Đưa ra danh sách nhân viên có lương lớn nhất theo từng phòng ban. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương, tên phòng ban.
4. Đưa ra lương lớn nhất theo từng phòng ban. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, lương.
5. Đưa ra những nhân viên có lương lớn nhất thuộc phòng 'Tổng hợp'. Thông tin đưa ra gồm mã nhân viên, họ tên, ngày sinh, lương.
6. Đưa ra phòng ban có nhân viên nhận lương lớn nhất. Thông tin đưa ra gồm tên phòng ban, mã nhân viên, tên nhân viên, lương.
7. Đếm số lượng nhân viên theo từng phòng ban. Thông tin đưa ra các phòng ban chưa có trưởng phòng.
8. Đưa ra các phòng ban có số lượng nhân viên nhiều nhất. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, số lượng nhân viên. (có thể có nhiều hơn một phòng ban có đông nhân viên nhất).
9. Đếm số lượng nhân viên là người quản lý theo từng phòng ban. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, số lượng nhân viên là quản lý.
10. Đưa ra phòng ban có nhiều nhân viên là người quản lý nhất.
11. Đưa ra danh sách nhân viên là người quản lý và số lượng nhân viên mà họ quản lý. Thông tin đưa ra gồm mã nhân viên, họ tên, tên phòng, số lượng nhân viên mà họ quản lý.
12. Đưa ra nhân viên quản lý nhiều nhân viên nhất. Thông tin đưa ra gồm mã nhân viên, họ tên, số lượng nhân viên mà họ quản lý.
13. Đưa ra số lượng người thân được nhận bảo hiểm theo từng nhân viên. Thông tin đưa ra gồm mã nhân viên, họ tên, số người thân.
14. Đưa ra nhân viên có nhiều người thân nhất. Thông tin đưa ra gồm mã nhân viên, họ tên, số lượng người thân.
15. Đưa ra nhân viên có nhiều người thân nhất. Thông tin đưa ra gồm mã nhân viên, họ tên, họ tên thân nhân.

16. Đếm số lượng đề án mã mỗi phòng ban phụ trách. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, số lượng đề án.
17. Đưa ra phòng ban phụ trách nhiều đề án nhất.
18. Đếm số lượng nhân viên tham gia cho từng đề án. Thông tin đưa ra gồm mã đề án, tên đề án, số lượng nhân viên.
19. Đưa ra đề án có số nhân viên tham gia đông nhất. Thông tin đưa ra gồm mã đề án, tên đề án, số lượng nhân viên.
20. Đưa ra đề án có ít nhân viên tham gia gia nhất.
21. Đưa ra nhân viên tham gia nhiều đề án nhất. Thông tin đưa ra gồm mã nhân viên, tên nhân viên, số đề án.
22. Đưa ra nhân viên tham gia tất cả các đề án.
23. Đưa ra nhân viên tham gia nhiều đề án nhất của mỗi phòng ban. Thông tin đưa ra gồm mã phòng ban, tên phòng ban, mã nhân viên, họ tên nhân viên, số lượng đề án.
24. Đưa ra danh sách nhân viên tham gia từ 2 đề án trở lên.
25. Tính tổng số giờ tham gia đề án của mỗi nhân viên. Danh sách được sắp xếp theo phòng ban.
26. Đưa ra phòng ban có nhân viên tham gia đề án với tổng số giờ nhiều nhất. Thông tin đưa ra gồm mã phòng, tên phòng, tổng số giờ.
27. Đưa ra phòng ban có nhân viên tham gia đề án với tổng số giờ nhiều nhất. Thông tin đưa ra gồm mã phòng, tên phòng, mã nhân viên, tên nhân viên tổng số giờ.
28. Đưa ra nhân viên có tổng số giờ tham gia đề án nhiều nhất. Thông tin đưa ra gồm mã nhân viên, tên nhân viên, tổng số giờ tham gia đề án.
29. Đưa ra danh sách nhân viên tham gia đề án có tổng số số giờ từ 10 giờ trở lên.
30. Đưa ra danh sách nhân viên tham gia đề án nhưng không có người thân nào.
31. Đưa ra danh sách nhân viên tham gia đề án có nhiều người thân nhất.

Bài 4: Thực hành theo đề tài nhóm. Thực hiện truy vấn dữ liệu với các hàm gom nhóm, kết hợp, truy vấn con.

Bài 5: Sinh viên thao tác với cơ sở dữ liệu quản lý đề án công ty, thực hiện các yêu cầu với hàm gom nhóm kết hợp.

Bài 6: Thực hành theo đề tài nhóm. Thực hiện truy vấn dữ liệu với các hàm gom nhóm, kết hợp, truy vấn con.

Nội dung thực tập 3: Lập trình T-SQL

Mục đích: Giúp sinh viên có được kỹ năng lập trình trên hệ quản trị cơ sở dữ liệu SQL Server.

Yêu cầu: Sinh viên thành thạo lập trình trên hệ quản trị cơ sở dữ liệu SQL Server, cụ thể lập trình với con trỏ, tạo thủ tục, hàm, trigger.

(Sinh viên đọc tài liệu tham khảo Giáo trình thực hành SQL từ trang 35-42, Slides bài giảng của Giáo viên: Chương 5 Lập trình T_SQL)

I. Tóm tắt lý thuyết:

1.Thủ tục: Sử dụng để thực hiện các nhiệm vụ hay công việc phức tạp

Có hai loại thủ tục lưu trữ:

- Thủ tục lưu hệ thống đề cập đến phương pháp quản trị dữ liệu và cập nhật thông tin vào các bảng (thường bắt đầu bằng sp_).
- Thủ tục lưu do người dùng định nghĩa.

Cú pháp:

CREATE PROC[EDURE] <tên thủ tục> [(<DSách tham số>)]
[WITH RECOMPILE|ENCRYPTION|RECOMPILE, ENCRYPTION]
AS
[DECLARE <biến cục bộ>
<Các câu lệnh của thủ tục>

- Lời gọi thủ tục có dạng:
  <tên_thủ_tục> [<danh_sách_các_đối_số>]

Số lượng các đối số và thứ tự của chúng phải phù hợp với số lượng và thứ tự của các tham số hình thức.

Trường hợp lời gọi thủ tục được thực hiện bên trong một thủ tục khác, bên trong một trigger hay kết hợp với các câu lệnh SQL khác, ta sử dụng cú pháp như sau:

EXEC[UTE] <tên_thủ_tục> [<danh_sách_các_đối_số>]

Ví dụ: EXECUTE MaxSLhang_200201

Thứ tự của các đối số được truyền cho thủ tục có thể không cần phải tuân theo thứ tự của các tham số như khi định nghĩa thủ tục nếu tất cả các đối số được viết dưới dạng:

@<tên_tham_số> = <giá_trị>

Ví dụ:

Thủ tục không có tham số: Tạo thủ tục xem lương lớn nhất, lương nhỏ nhất, tổng lương trong công ty

```
Create  proc sp_XemLuong
as
begin
 DECLARE @Ma CHAR(10),@LuongLN INT,@luongnn int, @tongluong int

SELECT @MA=Mapb from phongban where tenpb=@ten
SELECT @LuongLN=MAX(Luong), @luongnn=Min(luong), @tongluong=Sum(luong)
    FROM Nhanvien
    Print 'Luong LN là ' + convert(varchar(10),@luongLN)
   Print 'Luong NN là ' + convert(varchar(10),@luongnn)
   Print N'Tổng lương là ' + cast(@tongluong  as varchar)
end
```

Thực thi thủ tục:

sp_XemLuong

Ví dụ: Thủ tục có tham số

Tạo thủ tục xem lương lớn nhất, lương nhỏ nhất, tổng lương trong công ty cho phòng ban có tên là gì đó

```
Create  proc sp_XemLuong (@Ten nvarchar(50))
as
begin
 DECLARE @Ma CHAR(10),@LuongLN INT,@luongnn int, @tongluong int

SELECT @MA=Mapb from phongban where tenpb=@ten
SELECT @LuongLN=MAX(Luong), @luongnn=Min(luong), @tongluong=Sum(luong)
    FROM Nhanvien
    WHERE MaPB =@Ma
    Print 'Luong LN là ' + convert(varchar(10),@luongLN)
Print 'Luong NN là ' + convert(varchar(10),@luongnn)
Print N'Tổng lương là ' + cast(@tongluong  as varchar)
end
```

Ví dụ: Tạo thủ tục Thêm phòng ban:

```
CREATE PROC THEMPB(@MA  CHAR(10),@TEN NVARCHAR(50))
AS
 begin
  insert into phongban(mapb,tenpb)
  values (@ma,@tenpb)
 end
```
Hoặc
```
CREATE PROC THEMPB @MA  CHAR(10),@TEN NVARCHAR(50)
….
```
Thực hiện thủ tục: Thempb 'PB50',N'Kỹ thuật'

2. Hàm: Function

- Hàm là đối tượng cơ sở dữ liệu tương tự như thủ tục.
- Điểm khác biệt giữa hàm và thủ tục: Hàm trả về một giá trị thông qua tên hàm còn thủ tục thì không.
- Có thể sử dụng hàm như là một thành phần của một biểu thức (chẳng hạn, trong dsách chọn của lệnh SELECT).
- Có các hàm do HQT CSDL cung cấp sẵn
- Người sử dụng có thể định nghĩa các hàm nhằm phục vụ cho mục đích riêng của mình

Cú pháp:
```
CREATE FUNCTION tên_hàm ([danh_sách_tham_số]) RETURNS (kiểu_trả_về_của_hàm)
AS BEGIN
  các_câu_lệnh_của_hàm
END
```
Ví dụ:
```
CREATE FUNCTION thu(@ngay DATETIME) RETURNS NVARCHAR(10)
AS
   BEGIN
DECLARE @st NVARCHAR(10)
      SELECT @st=CASE DATEPART(DW,@ngay)
            WHEN 1 THEN 'Chu nhật'
            WHEN 2 THEN 'Thứ hai'
            WHEN 3 THEN 'Thứ ba'
            WHEN 4 THEN 'Thứ tư'
            WHEN 5 THEN 'Thứ năm'
            WHEN 6 THEN 'Thứ sáu'
            ELSE 'Thứ bảy' END
RETURN (@st)  /* Trị trả về của hàm */
   END
```
Lời gọi hàm:

- Sử dụng như hàm do hqt csdl cung cấp:
- SELECT masv,hodem,ten,dbo.thu(ngaysinh),ngaysinh
  FROM sinhvien
  WHERE malop='C24102'

Hàm với giá trị trả về là "dữ liệu kiểu bảng"
```
CREATE FUNCTION tên_hàm ([danh_sách_tham_số]) RETURNS TABLE
AS
RETURN (câu_lệnh_select)
```
Hàm với giá trị trả về là "dữ liệu kiểu bảng"

Các quy tắc:

- Kiểu trả về của hàm được chỉ định bởi mệnh đề RETURNS TABLE.

- Trong phần thân của hàm chỉ có duy nhất một câu lệnh RETURN xác định giá trị trả về của hàm thông qua duy nhất một câu lệnh SELECT (không sử dụng bất kỳ câu lệnh nào khác trong phần thân của hàm).

Ví dụ:

```
CREATE FUNCTION func_XemSV(@khoa SMALLINT) RETURNS TABLE
AS
        RETURN(SELECT masv,hodem,ten,ngaysinh
   FROM sinhvien INNER JOIN lop
          ON sinhvien.malop=lop.malop
   WHERE khoa=@khoa)
```

Lời gọi hàm:

- Để biết danh sách các sinh viên khoá 25, ta sử dụng câu lệnh như sau:

```
SELECT * FROM dbo.func_XemSV(25)
```

- Khi cần phải sử dụng nhiều câu lệnh trong phần thân hàm, cú pháp định nghĩa hàm:
- CREATE FUNCTION <*tên_hàm*>([<*danh_sách_tham_số*>]) RETURNS @<*biến_bảng*> TABLE <*định_nghĩa_bảng*>

```
AS
         BEGIN
                  <các_câu_lệnh_trong_thân_hàm>
         RETURN
 END
```

- Cấu trúc bảng trả về bởi hàm được xác định dựa vào định nghĩa của bảng trong mệnh đề RETURNS.
- Biến @<*biến_bảng*> trong mệnh đề RETURNS có phạm vi sử dụng trong hàm và được sử dụng như một tên bảng.
- Câu lệnh RETURN trong thân hàm không chỉ định giá trị trả về. Giá trị trả về của hàm chính là các dòng dữ liệu trong bảng có tên là @<*biếnbảng*> được định nghĩa trong mệnh đề RETURNS

```
CREATE FUNCTION Func_Tongsv(@khoa SMALLINT) RETURNS @bangthongke TABLE
   (
      makhoa   NVARCHAR(5),
      tenkhoa   NVARCHAR(50),
      tongsosv  INT
   ) AS
   BEGIN
      IF @khoa=0
         INSERT INTO @bangthongke
         SELECT khoa.makhoa,tenkhoa,COUNT(masv)
     FROM (khoa INNER JOIN lop
    ON khoa.makhoa=lop.makhoa)
                  INNER JOIN sinhvien on lop.malop=sinhvien.malop
    GROUP BY khoa.makhoa,tenkhoa
      ELSE
         INSERT INTO @bangthongke
         SELECT khoa.makhoa,tenkhoa,COUNT(masv)
       FROM (khoa INNER JOIN lop      ON khoa.makhoa=lop.makhoa)
                  INNER JOIN sinhvien ON lop.malop=sinhvien.malop
         WHERE khoa=@khoa
         GROUP BY khoa.makhoa,tenkhoa
      RETURN /*Trả kết quả về cho hàm*/
   END
```

Lời gọi hàm:

- SELECT * FROM dbo.func_TongSV(25)

  Cho kết quả thống kê tổng số sinh viên khoá 25 của mỗi khoa:

- SELECT * FROM dbo.func_TongSV(0)

  Cho kết quả thống kế tổng số sinh viên hiện có (tất cả các khoá) của mỗi khoa

Ví dụ: Thống kê số nhân viên tham gia dự án theo từng phòng ban

```
CREATE FUNCTION Func_TongNV RETURNS @bangthongke TABLE
  (
    madv      NCHAR(10),
    tendv     NVARCHAR(50),
    tongsonv  INT
  ) AS
  BEGIN
    INSERT INTO @bangthongke
      SELECT donvi.madv,tendv,COUNT(manv)
      FROM donvi,nhanvien,phancong
      WHERE donvi.madv=nhanvien.madv and
          nhanvien.manv=phancong.manv
      GROUP BY donvi.madv,tendv
  RETURN /*Trả kết quả về cho hàm*/
  END
```

3. Trigger:

là một kiểu thủ tục được kích hoạt tự động theo các sự kiện (events), nhằm:

- So sánh kiểu dữ liệu.
- Đọc dữ liệu từ các bảng nằm trong cơ sở dữ liệu khác.
- Thay đổi theo tầng hoặc xoá liên tục các bảng liên quan trong một cơ sở dữ liệu
- Huỷ bỏ các thay đổi không đúng
- Kiểm tra các ràng buộc phức tạp hơn việc bắt lỗi bằng ràng buộc CHECK

- Có 02 loại triggers:

  + Data Modification Language –DML (For | After triggers, Instead-of triggers)
  + Data Definition Language - DDL triggers (For | After triggers)

Cú pháp tạo DDL trigger:

CREATE TRIGGER trigger_name ON { ALL SERVER | DATABASE } [WITH [ENCRYPTION] | [EXECUTE AS CALLER | SELF | 'user_login'] { FOR | AFTER } { event_type | event_group } [,...n] AS { sql_statement [;] [...n] }

Execute As Caller là option mặc định.

Execute As User = 'user'

Cú pháp xóa trigger: DROP TRIGGER *trigger_name* [**,***...n*] ON { DATABASE | ALL SERVER } DISABLE TRIGGER { [*schema* .] *trigger_name* [**,***...n*] | ALL } ON {DATABASE | ALL SERVER } [;]

DDL triggers là các triggers được tự động gọi sau khi máy thực hiện các lệnh sau:

Create Table, Drop Table, Alter Procedure, Drop Schema, Create Login, …

- **Cú pháp với DML Trigger:**
  - Tạo CREATE TRIGGER <trigger_name> ON <table_name>|<view_name>

  [With encryption|EXECUTE AS { CALLER | SELF | '*user_name*' }]

  {[FOR| AFTER] [insert],[update],[delete] | Instead of}

  AS Transact-SQL statements
  - Xoá Drop Trigger <trg_name>
  - DISABLE TRIGGER { *trigger_name* [**,***...n*] | ALL } ON *object_name*

  ON: Chỉ ra rằng Trigger đang được viết cho bảng hoặc view nào.

  With encryption: nội dung của trigger sẽ được mã hóa.
- AFTER (FOR): các câu lệnh bên trong trigger sẽ được thực hiện sau khi các sự kiện tạo nên trigger đã xảy ra rồi.

- INSTEAD OF: sẽ bỏ qua sự kiện đã kích hoạt trigger mà thay vào đó sẽ thực hiện các dòng lệnh SQL bên trong Trigger

Các kiểu Trigger:

- Trigger Insert: Trigger được phát biểu bởi For insert. Trigger được thực hiện khi tiến hành thêm một mẫu tin vào bảng. Mẫu tin cần thêm sẽ được lưu trong một bảng tạm có tên là Inserted.
- Trigger Delete: Trigger được phát biểu bởi For delete. Trigger được thực hiện khi tiến hành xóa một mẫu tin trong bảng. Mẫu tin bị xóa sẽ được lưu trong một bảng tạm có tên là deleted.
- Trigger Update: Trigger được phát biểu bởi For update. Trigger được thực hiện khi tiến hành sửa một mẫu tin trong bảng. Mẫu tin bị thay đổi sẽ được lưu trong 2 bảng tạm có tên là Inserted (chứa giá trị mới) và Deleted (chứa giá trị cũ).

Chú ý:

- Trigger không thể được tạo ra trên bảng tạm thời hay bảng hệ thống. Trigger chỉ có thể được kích hoạt một cách tự động bởi một trong các event Insert, Update, Delete. Có thể áp dụng trigger cho View.
- Inserted và Deleted là 2 table tạm chỉ chứa trên bộ nhớ và chỉ có giá trị bên trong trigger mà thôi (nghĩa là chỉ nhìn thấy được trong trigger mà thôi). Ta có thể dùng thông tin trong 2 table này để so sánh dữ liệu cũ và mới hoặc kiểm tra xem dữ liệu mới.

Trigger dạng INSTEAD OF

Dạng INSTEAD OF sẽ bỏ qua sự kiện đã kích hoạt trigger mà thay vào đó sẽ thực hiện các dòng lệnh SQL bên trong Trigger

INSTEAD OF được chia làm 3 loại nhỏ: INSTEAD OF INSERT, INSTEAD OF UPDATE và INSTEAD OF DELETE.

Ví dụ:

Tạo trigger trên bảng nhanvien cho su kien

```
--insert, trigger thuc hien thong bao manv vua them
CREATE TRIGGER THEMNV ON NHANVIEN FOR INSERT
AS
 DECLARE @MA NCHAR(10)
BEGIN
 SELECT @MA=MANV FROM INSERTED
 PRINT 'Ma nhan vien vua them la '+@ma
END
```

Dạng INSTEAD OF sẽ bỏ qua sự kiện đã kích hoạt trigger mà thay vào đó sẽ thực hiện các dòng lệnh SQL bên trong Trigger

INSTEAD OF được chia làm 3 loại nhỏ:

INSTEAD OF INSERT, INSTEAD OF UPDATE và INSTEAD OF DELETE.

Ví dụ: Viết trigger để khi xóa nhân viên, thực hiện xóa thông tin tham gia dự án của nhân viên

```
Create TRIGGER XOANV ON NHANVIEN INSTEAD OF DELETE
AS
 DECLARE @MANV NCHAR(10)
BEGIN
        SELECT @MANV=MANV FROM DELETED

        DELETE PHANCONG
        WHERE  MANV = @MA

        DELETE NHANVIEN
        WHERE MANV=@MA
END
```

4. Con trỏ CurSor:

Dùng để chứa dữ liệu lấy từ CSDL,giống đối tượng recordset trong VB

Cú pháp khai báo biến Cursor :

Biến CURSOR
[phạm vi] [di chuyển][trạng thái][xử lý]
For câu lệnh Select
[For update [OF danh sách cột]]

- Phạm vi :
  - LoCal :chỉ sử dụng trong phạm vi khai báo(mặc định)
  - Global :sử dụng chung cho cả kết nối
- Di chuyển :
  - ForWard_Only :chỉ di chuyển một hướng từ trước ra sau(mặc định)
  - Scroll : di chuyển tùy ý
- Trạng thái
  - Static: dữ liệu trên Cursor không thay đổi mặc dù dữ liệu trong bảng nguồn thay đổi(mặc định)
  - Dynamic :dữ liệu trên Cursor sẽ thay đổi mặc dù dữ liệu trong bảng nguồn thay đổi
  - KeySet :giống Dynamic nhưng chỉ thay đổi những dòng bị cập nhật

Xử lý :

- Read_Only :chỉ đọc(mặc định)
- Scroll_Lock : đọc/ghi

- Câu lệnh select :không chứa các mệnh đề Into,Compute,Compute by
- Danh sách cột cập nhật : là danh sách các cột sẽ thay đổi được

❖ Mở CurSor: *Open tên_Biến_Cursor*
❖ Đọc và xử lý dữ liệu trong cursor

Cú pháp :

FETCH Hướng di chuyển From Tên_biến_Cursor Into Danh sách biến

Trong đó:

- Hướng di chuyển :
- NEXT: Di chuyển về sau
- PRIOR : Di chuyển về trước
- FIRST : Di chuyển về đầu
- LAST : Di chuyển về cuối
- ABSOLUTE n : di chuyển đến mẩu tin thứ n tính từ mẩu tin đầu tiên ,nếu n<0 : tính từ mẩu tin cuối
- RELATIVE n :di chuyển đến mẩu tin thứ n tính từ mẩu tin hiện hành

- Trong quá trình di chuyển để kiểm tra việc di chuyển có thành công hay không ta kiểm tra biến hệ thống @@FETCH_STATUS nếu <>0 thất bại

❖ Đóng Cursor :*Close @Tên_Biến*
❖ Giải phóng CurSor khỏi bộ nhớ :*DEALLOCATE @Tên_Biến*
❖ Chú ý :thứ tự các thao tác khi xử lý dữ liệu trên CurSor

1. Định nghĩa biến Cursor
2. Mở Cursor
3. Duyệt và xử lý dữ liệu trên Cursor
4. Đóng và giải phóng Cursur

Ví dụ 1:

Sử dụng con trỏ, cập nhật giá trị cho cột tổng số giờ trong bảng dự án (bằng tổng thời gian tham gia dự án của các nhân viên trong dự án tương ứng).

Hướng dẫn:

Trong bảng Duan thêm cột TongSG có cùng kiểu dữ liệu với cột SoGio trong bảng PhanCong.

Yêu cầu: Sử dụng con trỏ cập nhật dữ liệu cho cột TongSG

```
DECLARE @Ma NCHAR(10), @Tongsg Decimal(18,1)
 DECLARE cur_Pnhap CURSOR
```

```
 FORWARD_ONLY
 FOR
        SELECT MaDA
        FROM Duan
        OPEN cur_Pnhap
        WHILE 0=0
        BEGIN
              FETCH NEXT FROM cur_Pnhap
              INTO @Ma
              IF @@FETCH_STATUS<>0
                    BREAK
              SELECT @Tongsg = SUM(SOGIO)
FROM PhanCong WHERE SoDA=@Ma
        PRINT 'Đang cập nhật mã dự án:'
        PRINT @Ma

        UPDATE Duan
        SET TongSG = @Tongsg
        WHERE CURRENT OF cur_Pnhap
END
CLOSE cur_Pnhap
DEALLOCATE cur_Pnhap
```

B. Bài thực hành:

Bài 1: Thực hành với cơ sở dữ liệu Quản lý đề án công ty.

Trong bảng Dean thêm cột tổng thời gian.

Cập nhật giá trị cho cột tổng thời gian trong bảng đề án (bằng tổng thời gian tham gia đề án của các nhân viên trong đề án tương ứng).

Bài 2: Thực hành theo đề tài nhóm. Sử dụng con trỏ để cập nhật dữ liệu cho các trường thích hợp trong đề tài nhóm.

Bài 3: Thực hành với cơ sở dữ liệu Quản lý đề án công ty.

Thêm một đề án mới vào bảng Dean. Tên: Xây tòa cao ốc Thăng Long

Thực hiện thủ tục để bổ sung vào bảng Phancong tất cả các nhân viên thuộc phòng 'Triển khai dự án' tham gia vào đề án mới này

Bài 4: Thực hành theo đề tài nhóm. Sử dụng tủ tục để thao tác dữ liệu trong đề tài nhóm.

Bài 5: Thực hành với cơ sở dữ liệu Quản lý đề án công ty.

Tạo hàm thống kê số lượng đề án theo từng phòng ban.

Tạo hàm thống kê số lượng nhân viên tham gia đề án theo từng đề án.

Bài 6: Thực hành theo đề tài nhóm. Sử dụng hàm thao tác với cơ sở dữ liệu trong đề tài nhóm.

4. Bài thực hành Lập trình ứng dụng kết nối với cơ sở dữ liệu

KẾT NỐI CƠ SỞ DỮ LIỆU TRONG C#

-Bước 1: xác định chuỗi kết nối và câu lệnh SQL cần thực hiện

```
string strConn = @"Server=DOHUONG-PC\SQLEXPRESS;Initial Catalog=THUCHANH;Integrated Security=True";
```

hoặc:

```
connStr = @"Data Source= DOHUONG-PC\SQLEXPRESS;Initial Catalog=datagridviewdemo;User ID=sa;password=123456";
```

// chuỗi kết nối đến CSDL

connStr=@"Data Source=(local);Initial Catalog=THUCHANH;Integrated Security=True"

String sql = "select * from Table"// câu lệnh select cần thực hiện

-Bước 2: tạo đối tượng connection kết nối giữa ứng dụng và CSDL

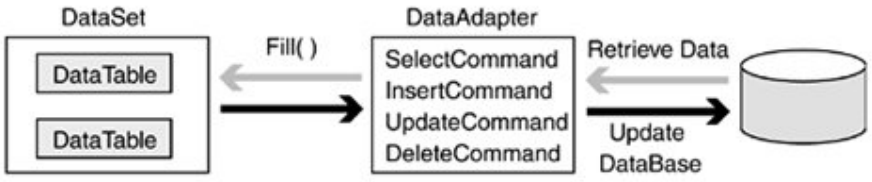
SqlConnection conn = new SqlConnection(); // khởi tạo một đối tượng kết nối

Conn.ConnectionString = connStr; // lấy đường dẫn đến cơ sở dữ liệu

Conn.Open(); // mở kết nối

-Bước 3 : tạo đối tượng SqlAdapter là cầu nối giữa dataset và datasource để thực hiện công việc như đọc hay cập nhật dữ liệu

SqlDataAdapter da = new SqlDataAdapter(sql,conn) ;



Bước 4 :dữ liệu đọc ra từ câu lệnh select được lưu vào 1 datatable trong dataset

DataTable dt = new DataTable() ; // khởi tạo đối tượng datatable

da.Fill(dt) ; // fill dữ aliệu vào datatable

Bước 5: nếu dữ liệu được hiển thị ra datagridview. Ta cần 1 DataView kết nối đến DataTable. Đối tượng DataView dùng cho việc sắp xếp,lọc, tìm kiếm…

DataView dv = new DataView(dt);

Bước 6: hiển thị dữ liệu lên datagridview

Dgr.DataSource = dv; // gán datasource cho datagridview

Dgr.AutoResizeColums(); // căn chỉnh lại chiều rộng các cột của datagridview

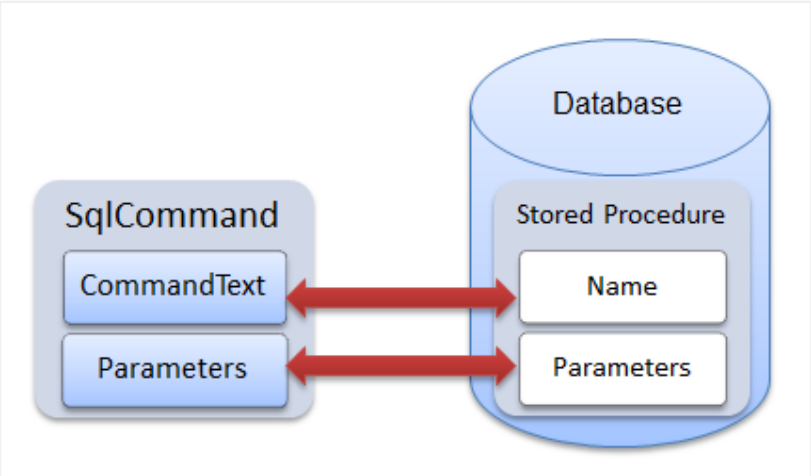
Bước 7: Đóng kết nối

conn.close();

SỬ DỤNG STORED PROCEDURE TRONG C#

Thiết lập đối tượng SqlCommand để sử dụng một stored procedure, ngoài ra biết được cách dùng các parameter với stored procedure.

Thay vì tạo các truy vấn động trong mã nguồn chương trình, ta có thể được lợi ích về việc tái sử dụng và hiệu suất khi sử dụng stored procedure.



Thực thi một Stored Procedure

Ngoài việc tạo các chuỗi lệnh SQL, ta phải thiết lập SqlCommand để thực thi stored procedure. Có hai bước để làm điều này: cho đối tượng SqlCommand biết stored procedure nào sẽ được thực thi và thiết lập chế độ thực thi stored procedure cho SqlCommand. Hai bước này được minh họa trong đoạn mã sau:

```
// 1.  create a command object identifying the stored procedure
SqlCommand cmd  = new SqlCommand(" Stored Procedure Name", conn);
 // 2. set the command object so it knows to execute a stored procedure
cmd.CommandType = CommandType.StoredProcedure;
```

Khi khai báo đối tượng SqlCommand trên, tham số đầu tiên được gán là "Stored Procedure Name". Đây là tên của stored procedure trong database SQL Server. Tham số thứ hai là đối tượng connection, tương tự như constructor của SqlCommand dùng để thực thi một câu truy vấn.

Dòng lệnh thứ hai chỉ cho đối tượng SqlCommand kiểu của lệnh sẽ được thực thi bằng cách gán property*CommandType* thành giá trị *StoredProcedure* của CommandType. Bằng cách thay đổi property CommandType này, SqlCommand sẽ hiểu được chuỗi lệnh trong tham số thứ nhất là một stored procedure. Phần còn lại của đoạn mã có thể được viết tương tự như các bài trước.

Truyền Parameter cho Stored Procedure

Dùng parameter cho stored procedure tương tự như dùng cho chuỗi lệnh truy vấn. Đoạn code sau cho thấy cách làm điều này:

```
1       // 1.  create a command object identifying the stored procedure
2       SqlCommand cmd  = new SqlCommand("Tên thủ tục trên SQLServer", conn);
3        // 2. set the command object so it knows  to execute a stored procedure
```

```
4       cmd.CommandType = CommandType.StoredProcedure;
5        // 3. add parameter to command, which will be passed to the stored procedure
6       cmd.Parameters.Add(new SqlParameter("@Tên tham số trong thủ tục", giá trị cần truyền));
```

Constructor của SqlCommand trên xác định tên của stored procedure, nếu thủ tục có tham số, ta phải tạo một parameter bằng cách dùng đối tượng SqlParameter. Tên của parameter được truyền trong tham số đầu tiên của SqlParameter constructor phải giống với tên của tham số của stored procedure. Sau đó thực thi command giống như với các đối tượng SqlCommand khác.

Tổng kết

Để thực thi stored procedure, ta cần chỉ ra tên của stored procedure trong tham số đầu tiên của một SqlCommand constructor và sau đó gán property *CommandType* của SqlCommand thành *StoredProcedured*. Ta cũng có thể truyền các tham số cho một stored procedure bằng cách dùng đối tượng SqlParameter, tương tự như cách làm với đối tượng SqlCommand dùng để thực thi một câu truy vấn.

VÍ DỤ MINH HỌA

Để giảm thiểu việc viết lệnh T-SQL trong mã code C#, người ta có thể tạo ra các thủ tục trong Hệ quản trị CSDL. Với cách này ta có thể dễ dàng bảo trì các mã T-SQL và Code C# trở lên ngắn gọn hơn. Đặc biệt là khi ta phải thực thi 1 thủ tục có thể lên đến hàng trang giấy hay vài trang giấy thì thực thi một thủ tục sẽ là giải pháp hữu hiệu trong lập trình với ADO.NET.

Ví dụ 1: Tạo form quản lý Phòng ban:

Giả sử, sử dụng CSDL SQL Server có tên QuanlyNhanvien, thực hiện trên bảng: **PhongBan**.

Bước 1: Thiết kế CSDL với bảng PhongBan:

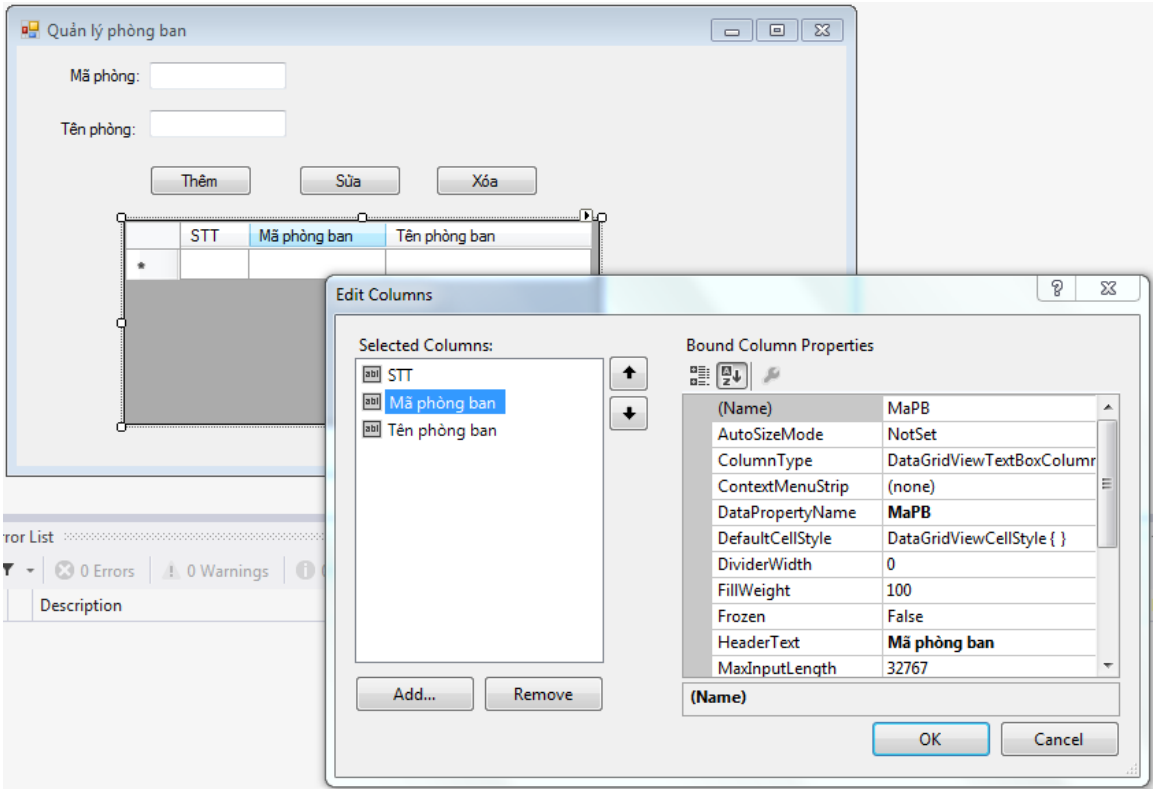
```
Create table PhongBan(MaPB char(10) primary key, TenPB nvarchar(50),DC nvarchar(100), SĐT varchar(50), MaTP char(10), NgayNC Date)
```

Bước 2: Viết các thủ tục cho phép thêm, sửa, xóa một phòng ban

```
/* Thủ tục thêm mới phòng ban*/
CREATE PROC SP_ThemPhongBan
(    @Ma char(10)
    @Ten nvarchar(50)
 )
AS
INSERT INTO PhongBan(MaPB, TenPB)
VALUES(@Ma,@Ten);

/* Thủ xóa một phòng ban*/
CREATE  SP_XoaPhongBan
(
```

```
    @Ma char(10)
)
AS
DELETE PhongBan
WHERE MaPB= @Ma;

/* Thủ tục sửa thông tin 1 phòng ban*/
CREATE PROC  SP_SuaPhongBan
(
    @Ma char(10),
    @Ten nvarchar(50)
    )
AS
UPDATE PhongBan
SET TenPB = @Ten
 WHERE MaPB= @Ma;
```

Bước 3: Sử dụng Visual Studio, chọn C#, ta cần thiết kế giao diện như sau:

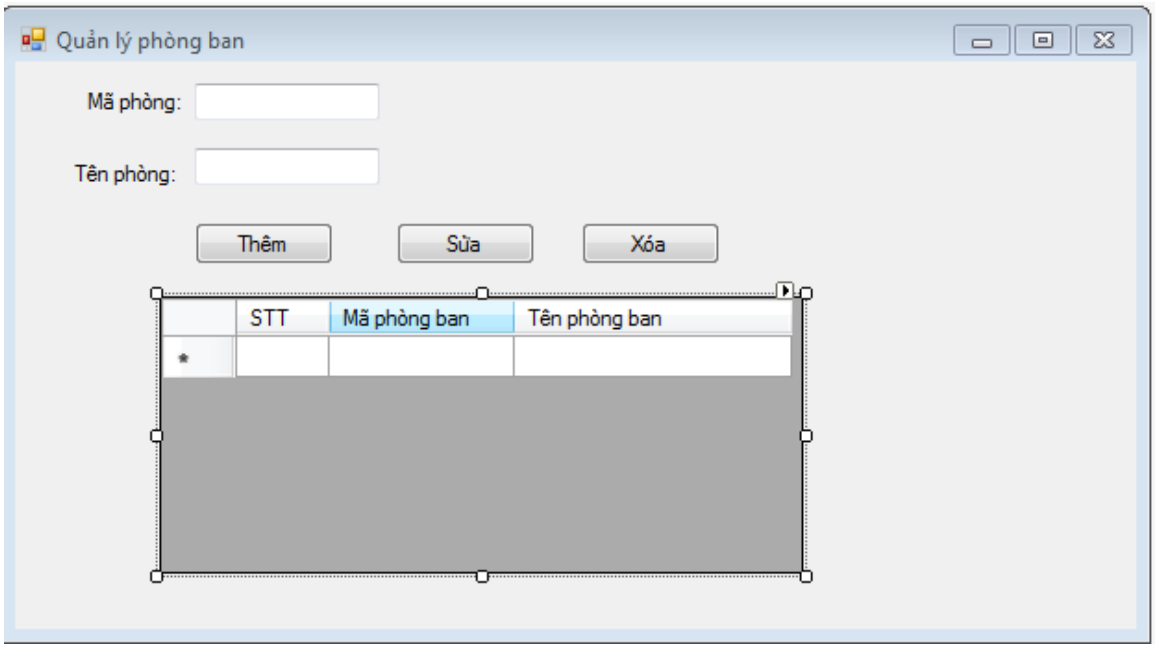


Trong ví dụ này ta ràng buộc dữ liệu DataGridView đặt tên là dgvPhongBan với 3 trường:

STT, MaPB, TenPB.

Sau đó bổ sung vào giao diện 2 ô TextBox (txtMa và txtTen), 3 Button: btnThem, btnSua, btnXoa.

Ta được giao diện như hình sau:



Bước 4: Lập trình hiển thị dữ liệu lên DataGridView

Đầu tiên ta khai báo và khởi tạo đối tượng Connection. Sau đó ta viết một hàm **LoadData**() dùng để load dữ liệu lên DataGridView vì hàm này còn được sử dụng lại khi ta thêm, sửa, xóa 1 bản ghi.

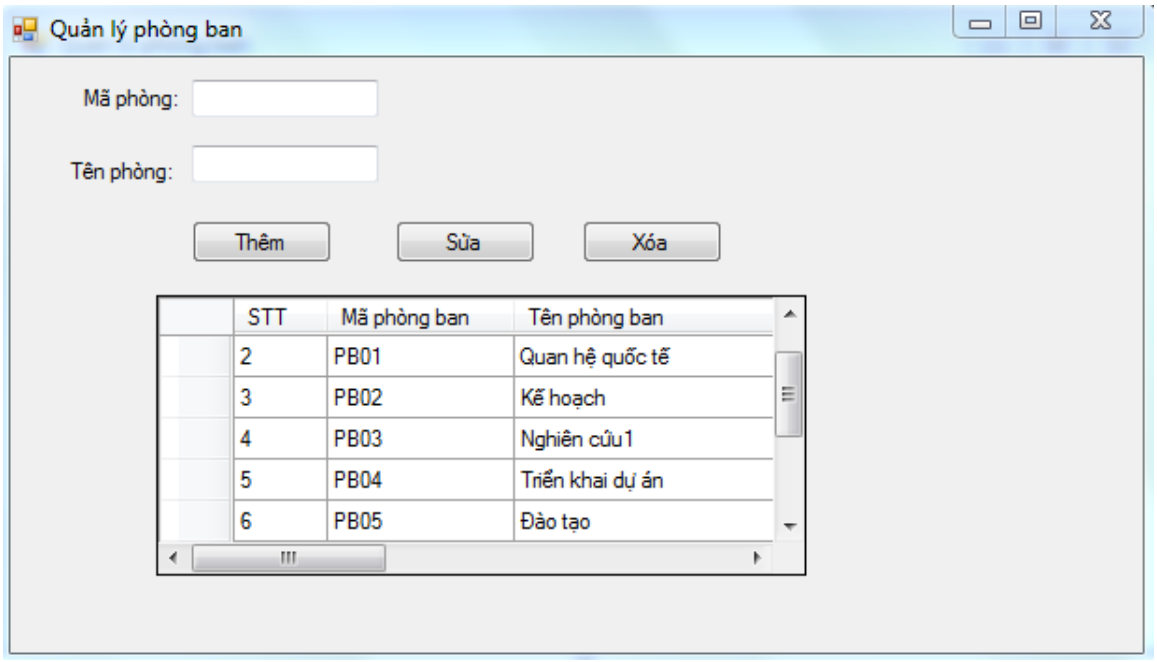
Viết code như sau:

```
string strConn = @"Server=.\SQLEXPRESS; Database=QuanlyNhanVien; Integrated Security=True";
SqlConnection conn;

private void LoadData()
    {
      SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM PhongBan", conn);
        DataTable dt = new DataTable();
        da.Fill(dt);

        dgvPhongBan.DataSource = dt;
    }

 private void frmPhongban_Load(object sender, EventArgs e)
    {
        conn = new SqlConnection(strConn);
        conn.Open();

        LoadData();
    }
```

Khi viết xong đoạn lệnh trên, chạy chương trình thì kết quả có thể đã hiển thị nhưng cột STT vẫn trống vì không có ràng buộc với trường này. Bởi vậy ta hãy viết trong sự kiện RowPrePaint của điều khiển DataGridView như sau:

```
private void dgvPhongBan_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
   {
      dgvPhongBan.Rows[e.RowIndex].Cells["STT"].Value = e.RowIndex + 1;
   }
```

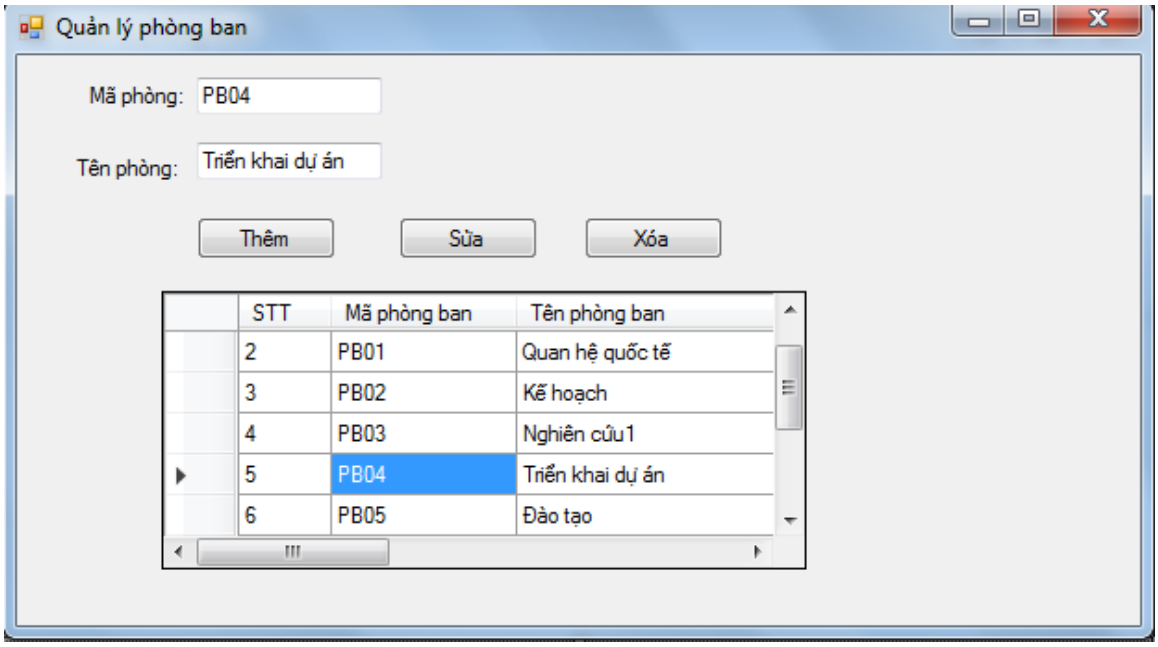
Lúc này cột số thứ tự sẽ điền số tự động như mong muốn.



Hãy chuyển sang bước 5.

Bước 5: Hiển thị dữ liệu lên TextBox tương ứng khi chọn 1 dòng trong DataGridView

```
private void dgvDeparts_CellClick(object sender, DataGridViewCellEventArgs e)
   {
      if (e.RowIndex >= 0 && e.ColumnIndex >= 0)
      {
      txtMa.Text = Convert.ToString(dgvPhongBan.CurrentRow.Cells["MaPB"].Value);

      txtTen.Text = Convert.ToString(dgvDeparts.CurrentRow.Cells["TenPB"].Value);
      }
   }
```



Bước 6: Thực thi thủ tục thêm mới một phòng ban: **SP_ThemPhong**.

```
private void btnThem_Click(object sender, EventArgs e)
   {
      // Khai báo và khởi tạo đối tượng Command, truyền vào tên thủ tục tương ứng
      SqlCommand cmd = new SqlCommand("SP_ThemPhongBan",conn);
      // Khai báo kiểu thực thi là Thực thi thủ tục
      cmd.CommandType = CommandType.StoredProcedure;
       // Khai báo và gán giá trị cho các tham số đầu vào của thủ tục
       // Khai báo tham số thứ nhất @Ma - là tên tham số được tạo trong thủ tục
      SqlParameter p = new SqlParameter("@Ma", txtMa.Text);
      cmd.Parameters.Add(p);
       // Khởi tạo tham số thứ 2 trong thủ tục là @Ten
      p = new SqlParameter("@Ten",txtTen.Text);
      cmd.Parameters.Add(p);
       // Thực thi thủ tục
      int count = cmd.ExecuteNonQuery();
       if (count > 0)
      {
         MessageBox.Show("Thêm mới thành công");
         LoadData();
      }
      else MessageBox.Show("Không thể thêm mới");
   }
```

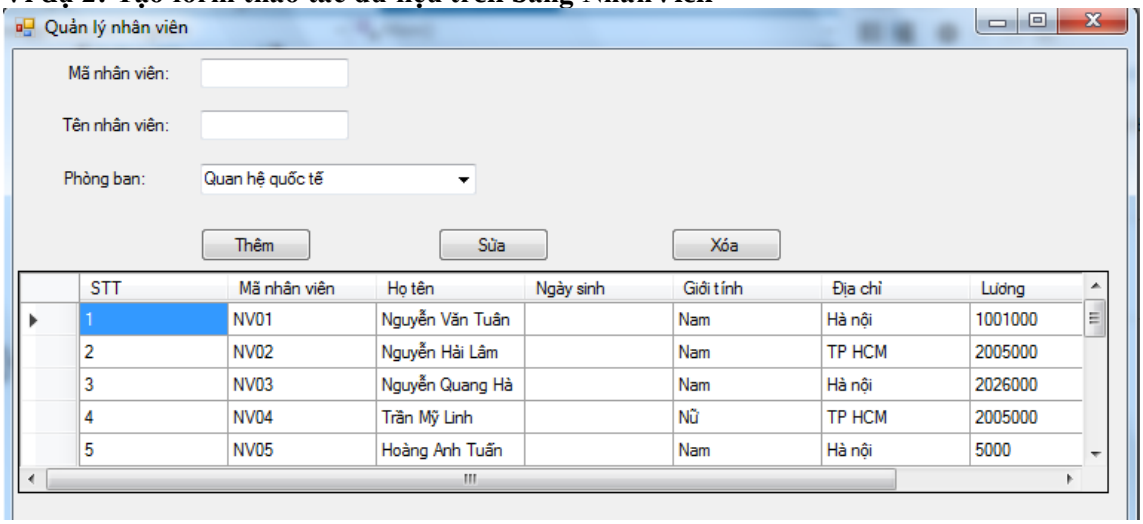
Có thể chạy chương trình và thử nghệm!

Bước 7: Tương tự cho việc thực thi các thủ tục sửa và xóa như sau:

```
private void btnSua_Click(object sender, EventArgs e)
    {
        SqlCommand cmd = new SqlCommand("SP_SuaPhongBan", conn);
        cmd.CommandType = CommandType.StoredProcedure;

       SqlParameter p = new SqlParameter("@Ma", txtMa.Text);
        cmd.Parameters.Add(p);
```

```
            p = new SqlParameter("@Ten", txtTen.Text);
            cmd.Parameters.Add(p);

            int count = cmd.ExecuteNonQuery();

            if (count > 0)
            {
                MessageBox.Show("Sửa thành công!");
                LoadData();
            }
            else MessageBox.Show("Không sửa được!");
        }


        private void btnXoa_Click(object sender, EventArgs e)
        {
if (MessageBox.Show("Bạn có chắc chắn muốn xóa không?", "Thông báo",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                SqlCommand cmd = new SqlCommand("SP_XoaPhongBan", conn);
                cmd.CommandType = CommandType.StoredProcedure;

                SqlParameter p = new SqlParameter("@Ma", txtMa.Text);
                cmd.Parameters.Add(p);

                int count = cmd.ExecuteNonQuery();

                if (count > 0)
                {
                    MessageBox.Show("Xóa thành công!");
                    LoadData();
                    txtMa.Text="";
                    txtTen.Text="";
                }
                else MessageBox.Show("Không thể xóa bản ghi hiện thời!");
            }
        }
```

Ví dụ 2: Tạo form thao tác dữ liệu trên bảng NhanVien



Tạo form frmNhanVien có 2 text box txtMaNV, txtTenNV, thêm comboBox tên là cboMaPB, 3 button lần lượt là btnThem,btnSua,btnXoa, 1 datagridview tên là dgvNhanVien

```
public partial class frmNhanVien : Form
    {
        SqlConnection conn;
        public frmNhanVien()
        {
            InitializeComponent();
            string strConn = "Server=DOHUONG-PC\\SQLEXPRESS;Initial
Catalog=THUCHANH;Integrated Security=True";
            conn = new SqlConnection(strConn);
            conn.Open();
            Show_CboPhongBan("");
            Show_Nhanvien("");

        }


        private void Show_CboPhongBan(string mapb)
        {
            string sql="Select mapb,tenpb From phongban";
            if (mapb != "")
                sql = sql + " Where mapb='" + mapb + "'";
            SqlDataAdapter daPhong = new SqlDataAdapter(sql, conn);
            DataTable dt = new DataTable();
            daPhong.Fill(dt);

            cboMaPB.DataSource = dt;
            cboMaPB.ValueMember = "MaPB";
            cboMaPB.DisplayMember = "TenPB";

       }
        private void Show_Nhanvien(string mapb)
        {
          string sql = "Select manv,hoten,gioitinh,diachi,luong,mapb
From nhanvien";
            if (mapb != "")
                sql = sql + " Where mapb='" + mapb+"'";
           SqlDataAdapter daNhanvien = new SqlDataAdapter(sql, conn);
            DataTable dt1 = new DataTable();

            daNhanvien.Fill(dt1);
            dgvNhanVien.DataSource = dt1;
        }
//Điền giá trị cho cột số thứ tự:
        private void dgvNhanVien_RowPrePaint(object sender,
DataGridViewRowPrePaintEventArgs e)
        {
            dgvNhanVien.Rows[e.RowIndex].Cells["STT"].Value = e.RowIndex +
1;
```

```
        }
//Lấy dữ liệu trên datagridview để hiển thị trên ô text box khi click đúp
chuột vào ô trên datagridview:
        private void dgvNhanVien_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (e.RowIndex >= 0 && e.ColumnIndex >= 0)
            {
                txtMaNV.Text =
(dgvNhanVien.CurrentRow.Cells["Ma"].Value.ToString());
                txtTenNV.Text =
Convert.ToString(dgvNhanVien.CurrentRow.Cells["Ten"].Value);

Show_CboPhongBan(dgvNhanVien.CurrentRow.Cells["MaPB"].Value.ToString());
            }
        }
//Show dữ liệu khi click vào combox Mã phòng ban
        private void cboMaPB_Click(object sender, EventArgs e)
        {
            Show_CboPhongBan("");
        }
//Viết lệnh cho nút lệnh Thêm

        private void btnThem_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("SP_ThemNV", conn);
            cmd.CommandType = CommandType.StoredProcedure;
             SqlParameter p = new SqlParameter("@Ma", txtMaNV.Text);
            cmd.Parameters.Add(p);
           p = new SqlParameter("@Ten", Convert.ToString(txtTenNV.Text));
            cmd.Parameters.Add(p);
            p = new SqlParameter("@MaPB",
cboMaPB.SelectedValue.ToString());
            cmd.Parameters.Add(p);
                // Thực thi thủ tục
            int count = cmd.ExecuteNonQuery();
            if (count > 0)
            {
                MessageBox.Show("Thêm mới thành công");
                Show_Nhanvien("");
            }
            else MessageBox.Show("Không thể thêm mới");

        }
//Viết lệnh cho nút lệnh Sửa

        private void btnSua_Click(object sender, EventArgs e)
        {
            SqlCommand cmd = new SqlCommand("SP_SuaNV", conn);
            cmd.CommandType = CommandType.StoredProcedure;
            SqlParameter p = new SqlParameter("@Ma", txtMaNV.Text);
            cmd.Parameters.Add(p);
            p = new SqlParameter("@Ten", Convert.ToString(txtTenNV.Text));
```

```
            cmd.Parameters.Add(p);
            p = new SqlParameter("@MaPB",
cboMaPB.SelectedValue.ToString());
            cmd.Parameters.Add(p);
            // Thực thi thủ tục
            int count = cmd.ExecuteNonQuery();
            if (count > 0)
            {
                MessageBox.Show("Sửa thành công");
                Show_Nhanvien("");
            }
            else MessageBox.Show("Không thể sửa được");

        }
//Viết lệnh cho nút lệnh Xóa

        private void btnXoa_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("Bạn có chắc chắn muốn xóa Nhân viên  ?" +
txtMaNV.Text, "Thông báo", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) == DialogResult.Yes)
            {
                SqlCommand cmd = new SqlCommand("SP_XoaNV", conn);
                cmd.CommandType = CommandType.StoredProcedure;

              SqlParameter p = new SqlParameter("@Ma", txtMaNV.Text);
                cmd.Parameters.Add(p);
                int count = cmd.ExecuteNonQuery();
           if (count > 0)
                {
                    MessageBox.Show("Xóa thành công!");
                    Show_Nhanvien("");

                }
           else MessageBox.Show("Không thể xóa bản ghi hiện thời!");
            }

        }

    }
```

Bài 3: Sinh viên xây dựng ứng dụng trên đề tài nhóm. Thực hiện kết nối với cơ sở dữ liệu trong đề tài nhóm. Xây dựng các giao diện Thêm, Sửa, Xóa, Tìm kiếm.