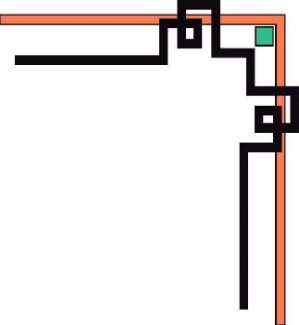
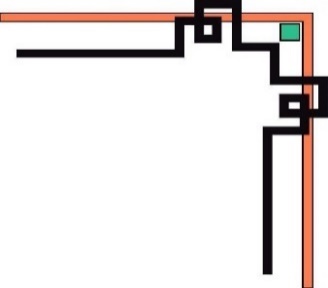
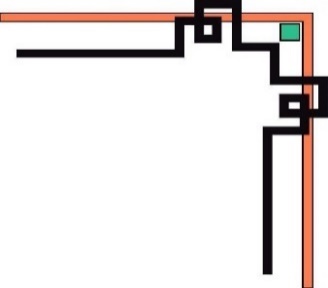
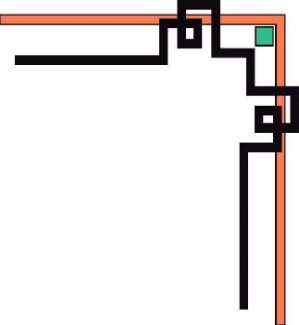
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**



-----🙞🙜🕮🙞🙜-----

****

**Báo Cáo Cuối Kì**

**Môn Học : CSDL Phân Tán**

**Đề Tài :**

**Quản Lí Cửa Hàng Bán Điện Thoại**

**Giáo Viên Hướng Dẫn : Ths. Lê Minh Châu**

**Nhóm Thực Hiện : Lê Thiện Đức**

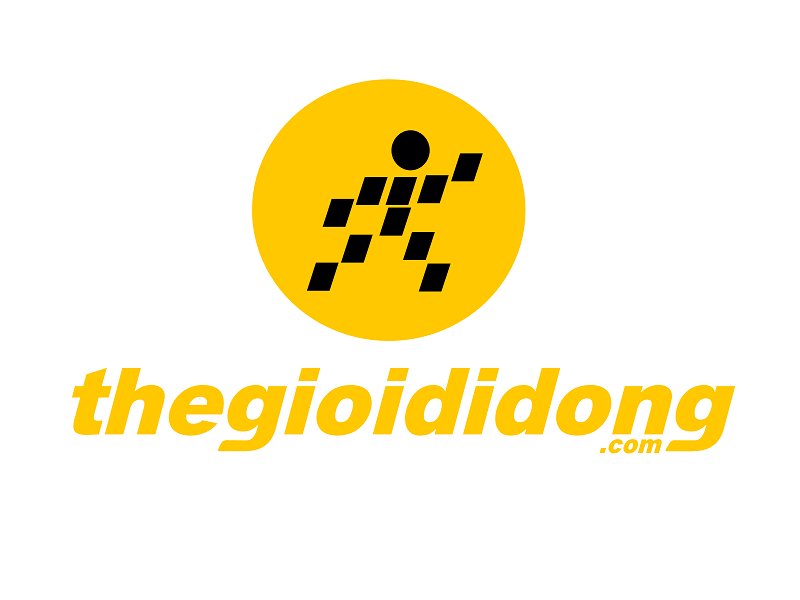
**Vũ Văn Bảo**

**TP-HỒ CHÍ MINH – THÁNG 6/2019**

# PHÁT BIỂU BÀI TOÁN

## GIỚI THIỆU VỀ CÔNG TY THẾ GIỚI DI ĐỘNG

### Giới thiệu sơ lược

****

Hình 1. Logo thế giới di động

Công ty TNHH Thế Giới Di Động[2] là một trong những doanh nghiệp hàng đầu Việt Nam trong lĩnh vực bán lẻ điện thoại di động và các sản phẩm điện tử. Công ty được thành lập vào tháng 03/2004 bởi 5 người bạn là Trần Lê Quân, Nguyễn Đức Tài, Đinh Anh Huân, Điêu Chính Hải Triều và Trần Huy Thanh Tùng. Công ty có trụ sở chính tại Tòa nhà MWG – Lô T2 – 1.2, Đường D1, Khu Công Nghệ Cao, Phường Tân Phú, Quận 9, TP. Hồ Chí Minh và gồm nhiều thành viên khác nhau, bao gồm:

- Công ty Cổ phần Thế giới di động (www.thegioididong.com )

- Công ty Cổ phần Thế giới điện tử (www.dienmayxanh.com )

- Siêu thị Bách Hoá Xanh (www.bachhoaxanh.com )

- Nhà thuốc An Khang ( www.nhathuocankhang.com )

- Thương hiệu Bluetronics ( www.bluetronics.com )

- Thương hiệu 4KFarm ( www.4kfarm.com )

Người sáng lập và Chủ tịch HĐQT của Công ty cổ phần Thế Giới Di Động là ông Nguyễn Đức Tài. Ngoài ra, công ty còn có một số nhân vật quan trọng khác trong bộ máy quản lý, như:

- Ông Trần Kinh Doanh – Tổng Giám đốc kiêm Giám đốc phát triển kinh doanh

- Ông Đặng Minh Lượm – Giám đốc nhân sự

- Ông Đinh Anh Huân – Giám đốc Dienmayxanh.com

- Ông Đoàn Văn Hiểu Em – Đại diện Pháp luật

### Đặc điểm kinh doanh

Thegioididong.com là một trong những thương hiệu hàng đầu của Công ty Cổ phần Thế giới di động (Mobile World JSC), có mã Chứng Khoán là MWG[2]. Đây là một tập đoàn bán lẻ uy tín tại Việt Nam, hoạt động trong các lĩnh vực như điện thoại di động, thiết bị kỹ thuật số, thương mại điện tử và bán lẻ thực phẩm cùng nhu yếu phẩm. Theo nghiên cứu của EMPEA, Thế giới di động đã chiếm 25% thị phần bán lẻ điện thoại di động và trở thành doanh nghiệp lớn nhất trong lĩnh vực này ở Việt Nam vào năm 2014.  
Thế giới di động không chỉ có mạng lưới cửa hàng rộng khắp mà còn có phong cách bán hàng tư vấn chuyên nghiệp và hiệu quả thông qua các trang web như www.thegioididong.com, www.dienmayxanh.com , www.bachhoaxanh.com ,... Khách hàng khi mua hàng qua trang web sẽ được hưởng ưu đãi giảm 5% so với giá bán tại cửa hàng. Ngoài ra, Thế giới di động còn được biết đến với các chính sách bán hàng và chăm sóc khách hàng thiết thực và hấp dẫn, dựa trên lợi ích của khách hàng. Thế giới di động cũng là đơn vị tiên phong về chất lượng sản phẩm và dịch vụ, được nhiều người tiêu dùng và đối tác tin tưởng và bình chọn.

### Thị trường hoạt động

Thế giới di động có mạng lưới cửa hàng rộng khắp Việt Nam[3], với nhiều chi nhánh ở mỗi tỉnh, mỗi thành phố. Tổng số cửa hàng trên toàn quốc là hơn 4500, bao gồm:

* 2173 cửa hàng Thế Giới Di Động
* 1224 cửa hàng Điện Máy Xanh
* 1661 cửa hàng Bách Hóa Xanh

Thế giới di động phục vụ khoảng 520000 khách hàng mỗi ngày và tiếp tục mở rộng quy mô kinh doanh. Các hệ thống siêu thị của Thế giới di động không chỉ phát triển trên thực địa mà còn đầu tư vào các website chuyên nghiệp. Nhờ vậy, Thế giới di động đã tạo được uy tín và thương hiệu trong thị trường thương mại Việt Nam.

### Tiên phong trong lĩnh vực thương mại điện tử

Công ty Thế giới di động không chỉ là nhà bán lẻ số một về doanh thu và lợi nhuận, mà còn là một trong những công ty tiên phong trong lĩnh vực thương mại điện tử tại Việt Nam [2] [4]. Công ty đã xây dựng được một hệ thống website, ứng dụng và dịch vụ trực tuyến chất lượng cao, phục vụ nhu cầu của hàng triệu khách hàng mỗi ngày. Công ty cũng đã áp dụng nhiều công nghệ mới như trí tuệ nhân tạo, blockchain và big data để nâng cao hiệu quả kinh doanh và trải nghiệm khách hàng.

## ĐẶT VẤN ĐỀ

Hiện tại, công ty đang sử dụng một hệ thống cơ sở dữ liệu tập trung. Tuy nhiên, với quy mô ngày càng mở rộng, việc duy trì và nâng cấp hệ thống này đang gặp nhiều khó khăn. Đặc biệt, việc đảm bảo tính nhất quán, khả năng chịu lỗi và hiệu suất của hệ thống là một thách thức lớn.

Mục tiêu của dự án này là thiết kế và triển khai một hệ thống cơ sở dữ liệu phân tán cho công ty Thế Giới Di Động. Hệ thống mới này cần phải đáp ứng được các yêu cầu về hiệu suất, khả năng mở rộng, độ tin cậy và bảo mật.

Khi khách hàng có yêu cầu mua các sản phẩm của cửa hàng thì hệ thống phải cung cấp sản phẩm kịp thời, chính xác cho khách hàng.

Khi cửa hàng muốn nhập sản phẩm về kho thì hệ thống nhận hàng đầy đủ các sản phẩm đã đặt.

Việc thực hiện các quy trình trên không để xảy ra các sai sót về sản phẩm nhận cũng như sản phẩm cung cấp và cả quá trình thanh toán.

Để đạt được mục tiêu này, chúng tôi sẽ tiến hành nghiên cứu các công nghệ và phương pháp thiết kế cơ sở dữ liệu phân tán hiện đại. Chúng tôi cũng sẽ phân tích chi tiết các yêu cầu của công ty để đảm bảo rằng hệ thống mới sẽ phù hợp với nhu cầu thực tế.

### Tổ Chức Cửa Hàng.

Giả sử hệ thống bán hàng của Thế Giới Di Động có 2 chi nhánh cửa hàng ở Hà Nội và Thành phố Hồ Chí Minh. Mỗi chi nhánh có 1 người quản lí và 2-3 nhân viên bán hàng. Đối tượng khách hàng chủ yếu là nhân viên văn phòng, doanh nhân và học sinh sinh viên đang có nhu cầu về điện thoại.

**Các hoạt động chính:** bán hàng, lưu trữ các dữ liệu liên quan, quản lí điện thoại, nhân viên, quản lí thông tin khách hàng, hóa đơn, hãng cung cấp điện thoại.

Việc quản lí cửa hàng điện thoại được phân cấp theo từng bộ phận :

* Quản lí chi nhánh.
* Quản lí nhân viên.
* Quản lí nhập điện thoại.
* Quản lí sản phẩm còn trong kho.
* Nhân viên bán hàng.
* Lập hóa đơn bán hàng.
* Quản lí thông tin khách hàng.
* Xuất hóa đơn, thu/nhận tiền.
* Quy trình nhập điện thoại.
* Quy trình bán điện thoại.
* Khi khách hàng tới quầy thanh toán, nhân viên bán hàng sẽ quét mã vạch đối với từng điện thoại, in hóa đơn cho khách hàng và thu nhận tiền từ khách hàng.

### Cơ cấu tổ chức cửa hàng tại mỗi chi nhánh

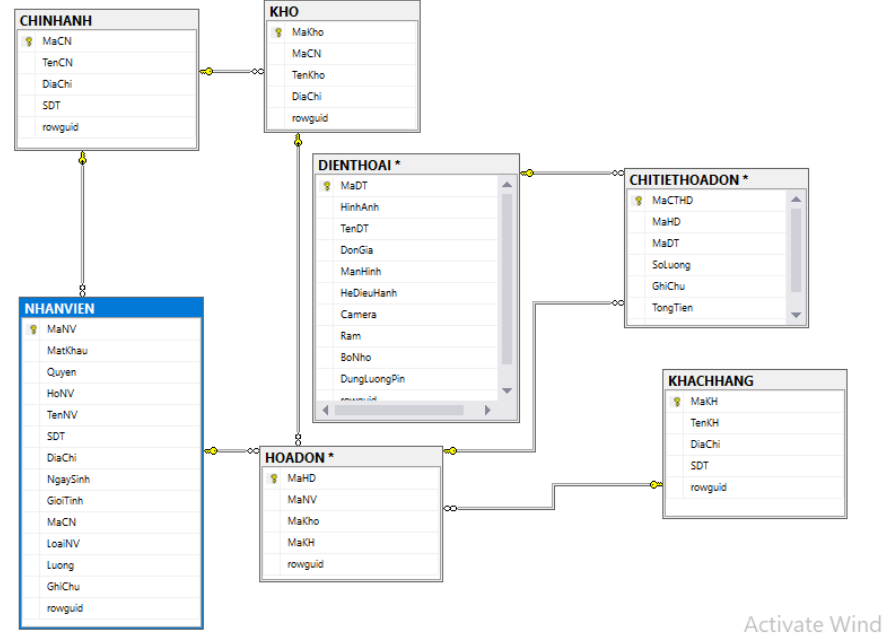
Một chi nhánh bán điện thoại di động có 2 đội, hoạt động tương đối độc lập với nhau:

Đội 1: Đảm nhận việc đặt hàng của khách hàng dựa trên yêu cầu mua sản phẩm của khách hàng đối với cửa hàng cũng như đảm nhận việc nhập hàng về kho cho cửa hàng. Đội này sử dụng một máy tính nhỏ, trên đó hoạt động theo một chương trình được cài đặt là hệ Đặt Hàng, trợ giúp các việc lập hóa đơn các thông tin về khách hàng cũng như sản phẩm và theo dõi sự hoàn tất trong quá trình giao dịch của đơn hàng.

Đội 2: Đảm nhận việc ghi nhận hàng về kho và cung cấp sản phẩm cho khách hàng. Đội này cũng có 1 máy tính nhỏ, trên đó có 1 hệ chương trình hệ Nhận Phát Hàng trợ giúp các việc ghi nhận hàng về kho và làm thủ tục cung cấp sản phẩm cho khách hàng.

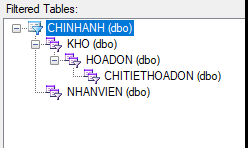
# THIẾT KẾ VÀ CÀI ĐẶT CƠ SỞ DỮ LIỆU

## Lược đồ toàn cục



Lược đồ toàn cục CSDL

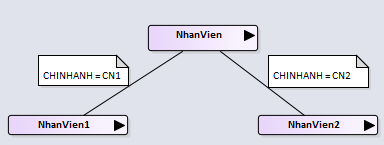
## Lược đồ phân mảnh.



Lược đồ phân mảnh

## Phân mảnh.

* Quan hệ NHANVIEN(nhân viên) được phân ra làm 2 mảnh ngang chính theo từng chi nhánh:
  + Nhân viên chi nhánh Hà Nội.
  + Nhân viên chi nhánh HCM.



Phân mảnh nhân viên theo từng chi nhánh

NhanVien1 = бChiNhanh = CN1 NhanVien

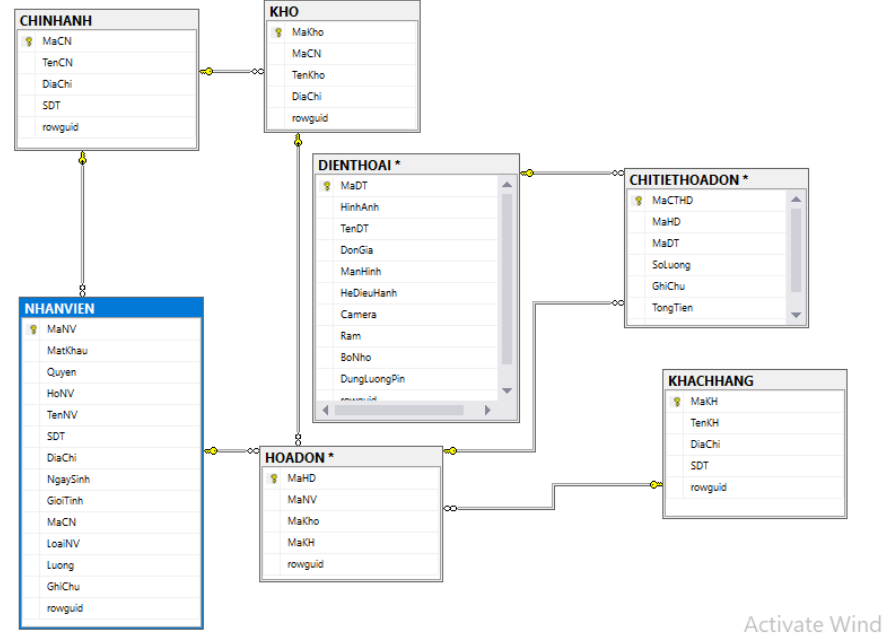
NhanVien2 = бChiNhanh = CN2 NhanVien

NhanVien = NhanVien1 Ս NhanVien2

* Tương tự với phân mảnh KHO (kho).

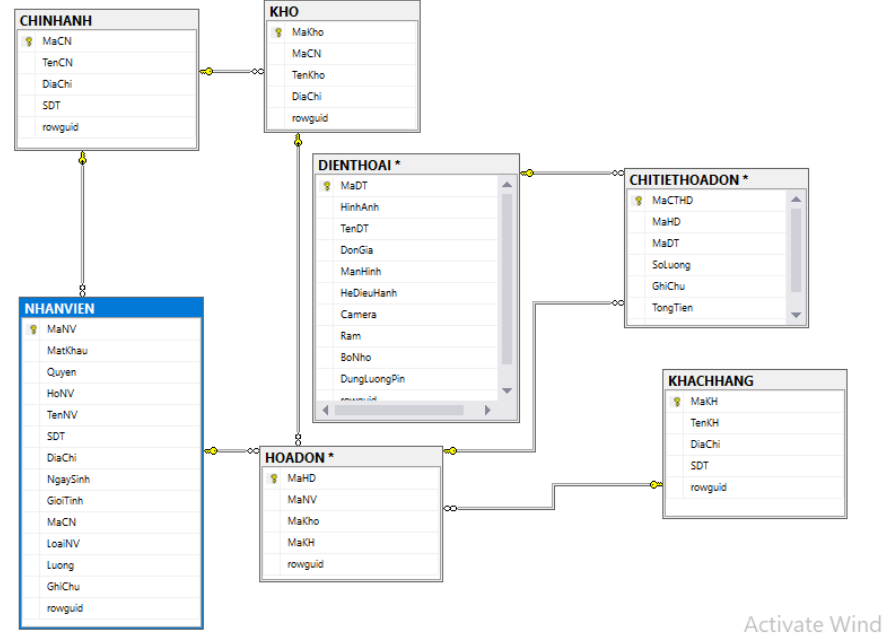
## Lược đồ từng chi nhánh

### Chi nhánh Hà Nội



Lược đồ tổng thể chi nhánh Hà Nội

### Chi nhánh thành phố Hồ Chí Minh



Lược đồ tổng thể chi nhánh thành phố Hồ Chí Minh

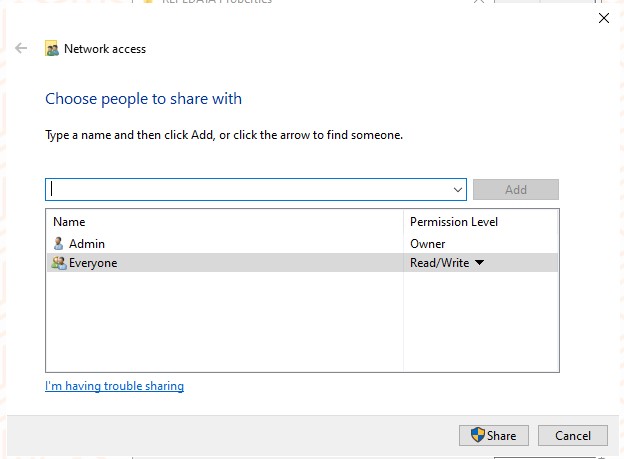
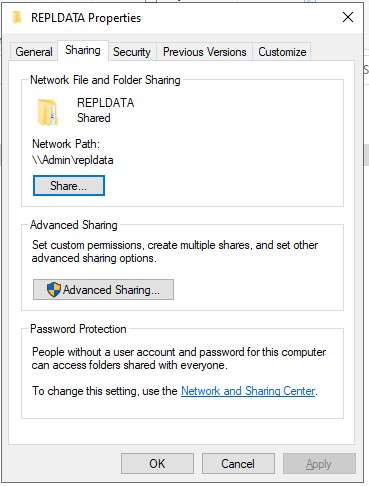
# CÀI ĐẠT CSDL PHÂN TÁN

## CÀI ĐẶT

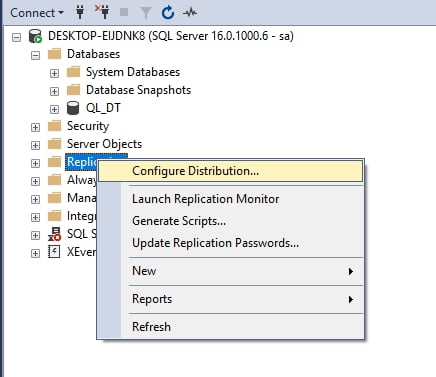
### Cấu hình hệ thống

Cài 3 server SQL Server 2019 bản Developer.

### Cấu hình Distribution



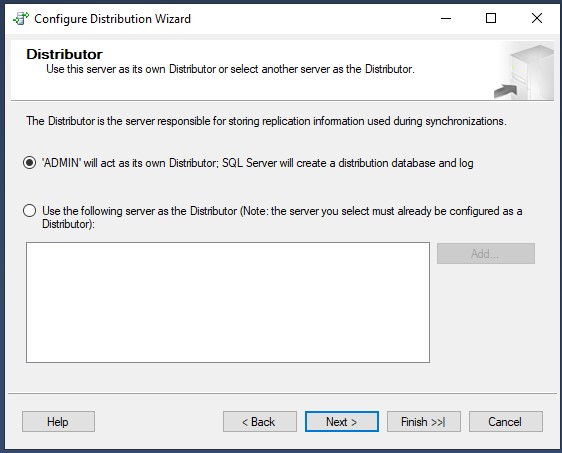
Hình 2.1. Chia sẻ thư mục snapshot



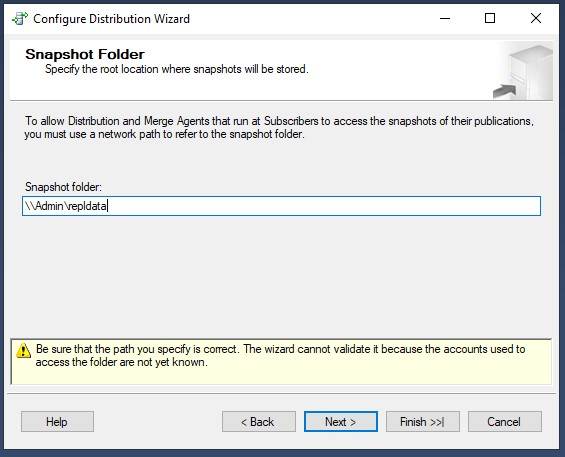
*Hình 2.2. Trên server distributer: Right click vào Mục repication chọn Config*



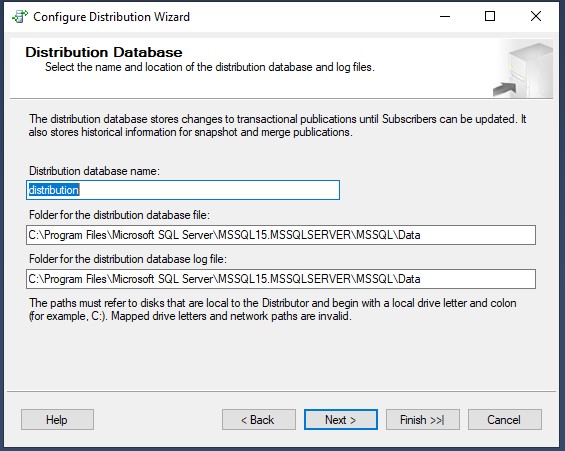
*Hình 2.3. Chọn next*



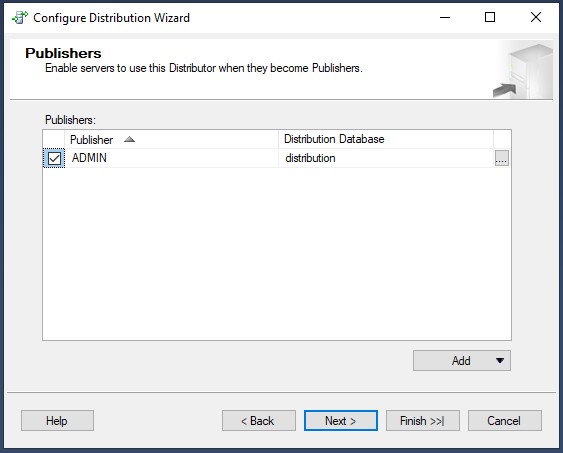
*Hình 2.4. Để mặc định và chọn next*



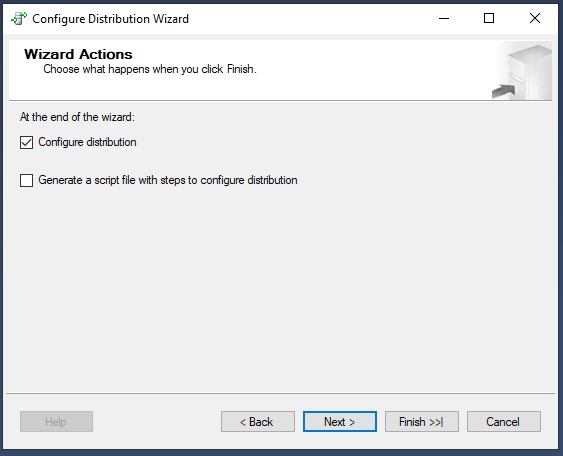
*Hình 2.5. Chọn thư mục chứa file snapshot và chọn next*



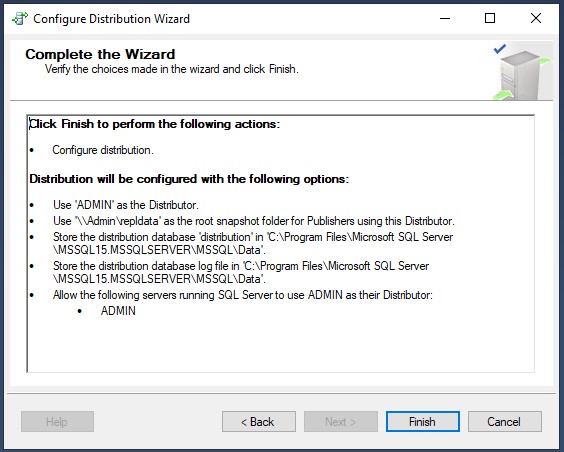
*Hình 2.6. Để mặc định và chọn next*



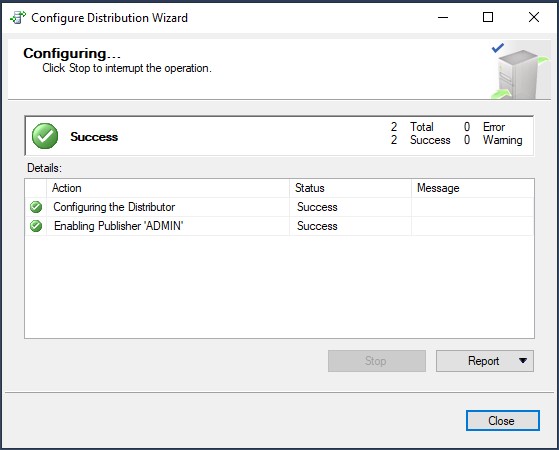
*Hình 2.7. Để mặc định và chọn next*



*Hình 2.8. Để mặc định và chọn next*

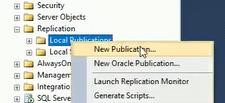


*Hình 2.9. Chọn Finish và chờ kết quả*

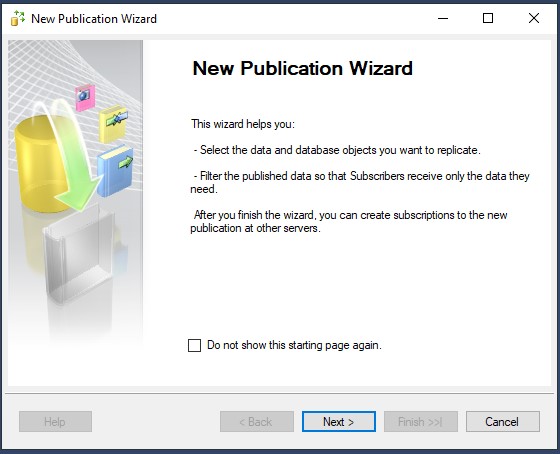


*Hình 2.10. Kết quả cài đặt thành công*

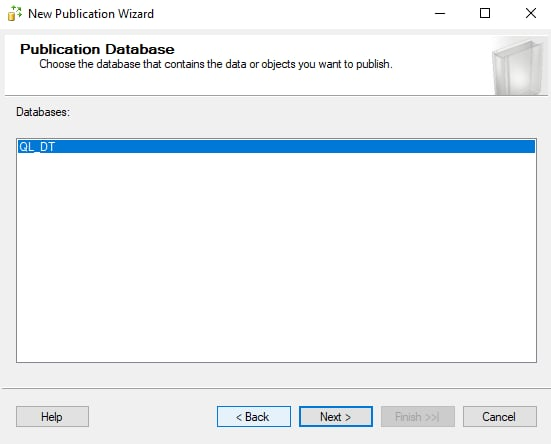
### Tạo Publication



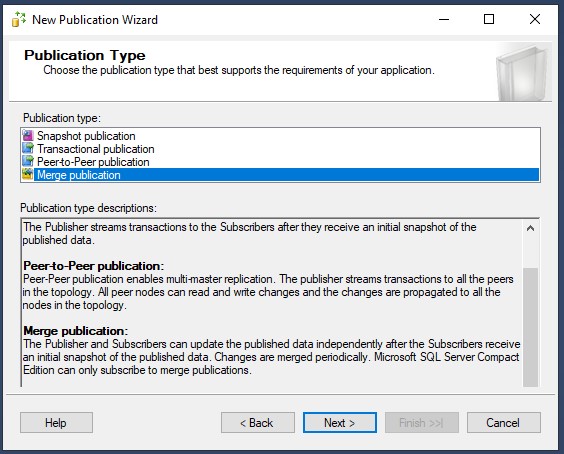
*Hình 2.12. Vào Publisher server, trong mục Local Publication chọn New Publication..*



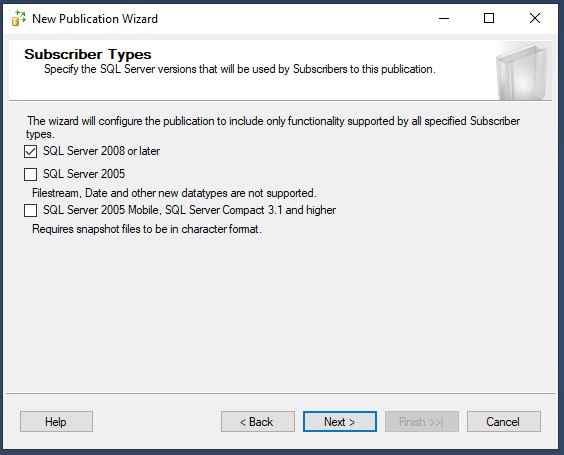
*Hình 2.13. Chọn next*



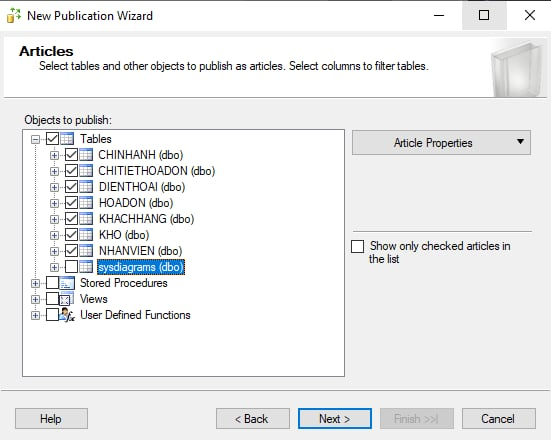
*Hình 2.14. Chọn Database cần Publication*



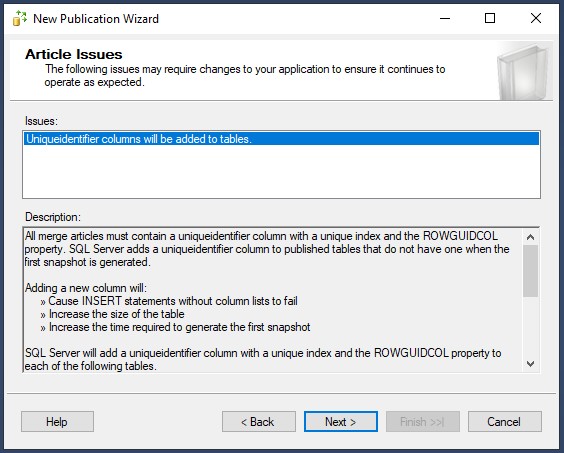
*Hình 2.15. Chọn Merge publication và next*



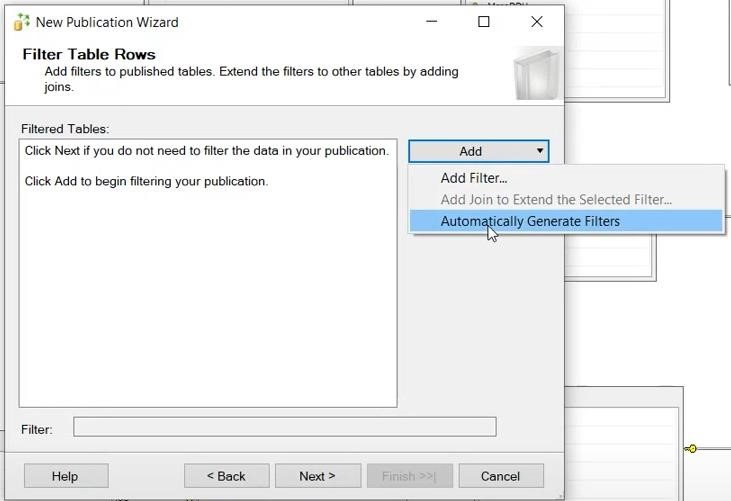
*Hình 2.16. Để mặc định và chọn next*



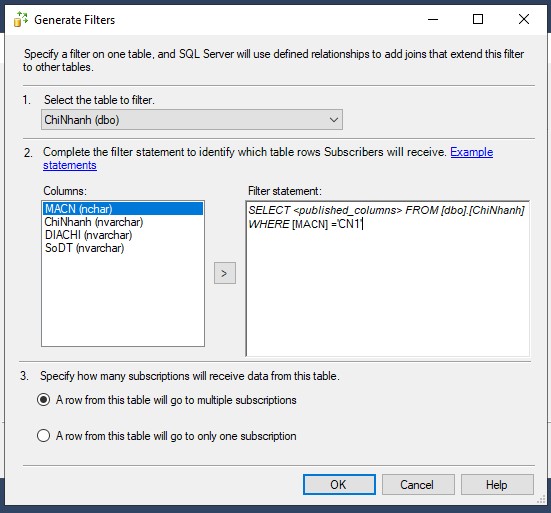
*Hình 2.17. Chọn các tables cần thiết và lưu ý không chọn tables sysdiagrams*



Hình 2.18. Chọn next



*Hình 2.19. Chọn để add các Filter Table*

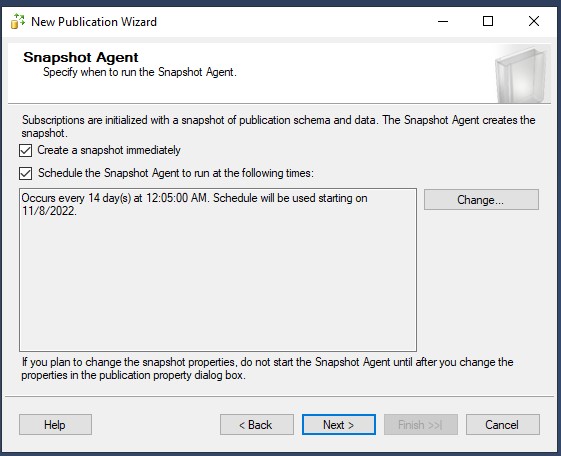


*Hình 2.20. Thêm điều kiện*

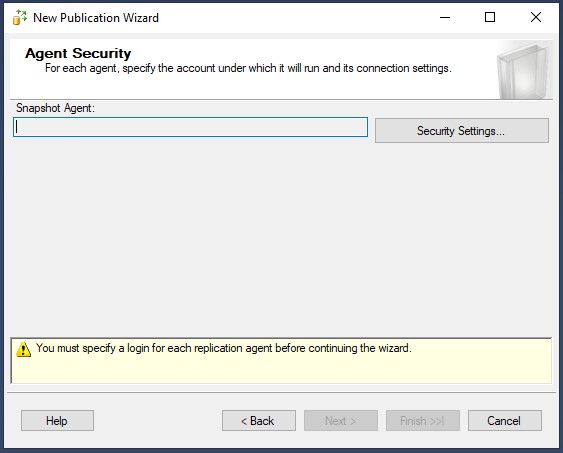
A screenshot of a computer

Description automatically generated

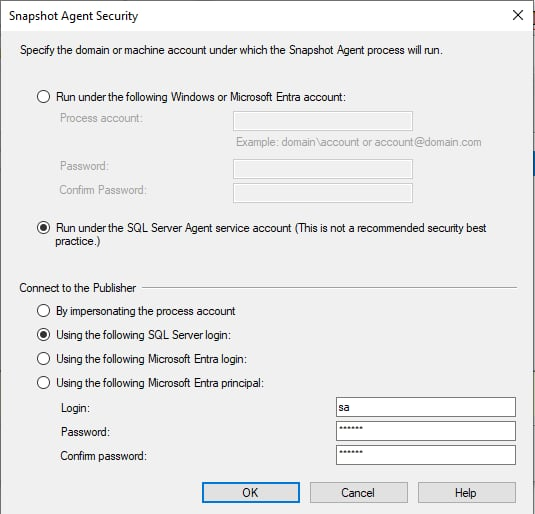
*Hình 2.21. Kiểm tra các table*



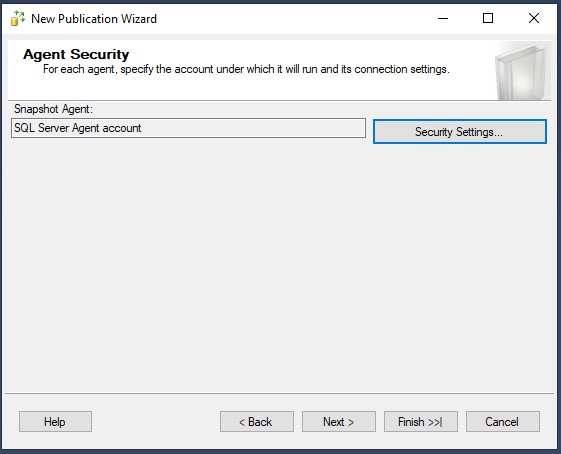
*Hình 2.22. Để mặc định và chọn next*



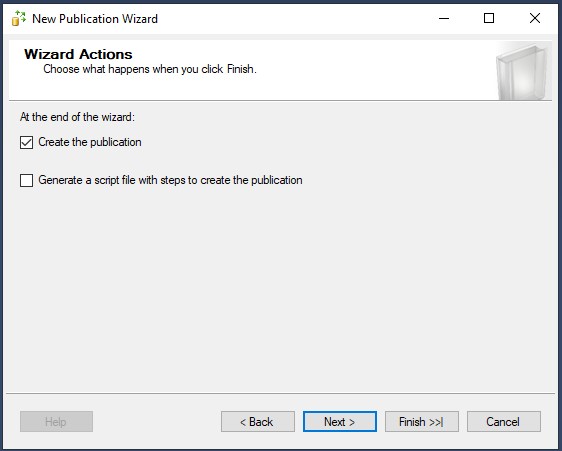
*Hình 2.23. Click Security Settings*



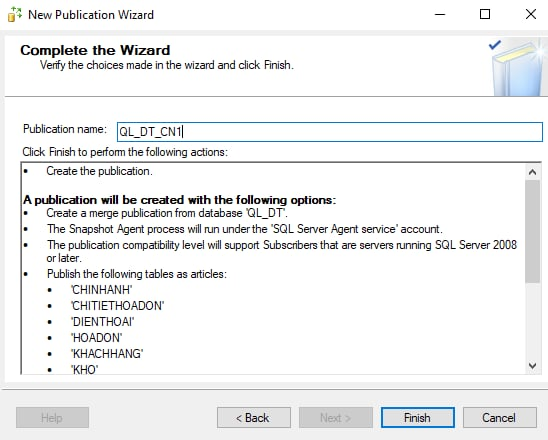
*Hình 2.24. Click chọn và nhấn OK*



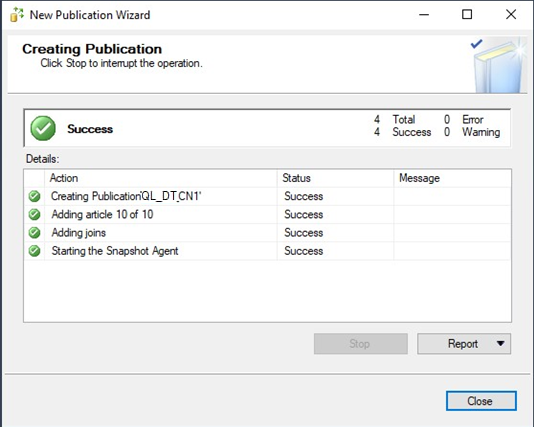
*Hình 2.25. Chọn next*



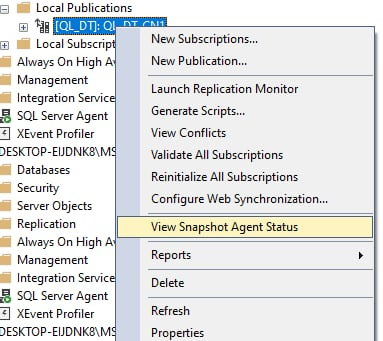
*Hình 2.26. Chọn next*



*Hình 2.25. Chọn finish và chờ kết quả*



*Hình 2.26. Hiển thị kết quả cài đặt*

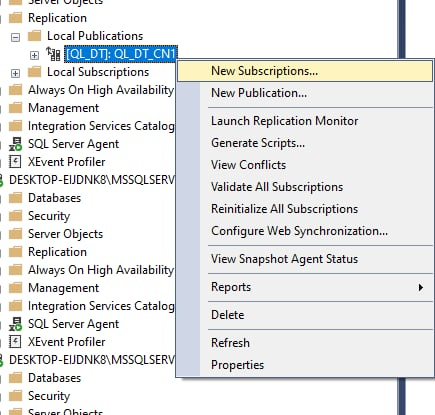


*Hình 2.27. Vào mục Local Publication trong mục QL\_DT\_CN1 chuột phải chọn View Snapshot Agent Status..*

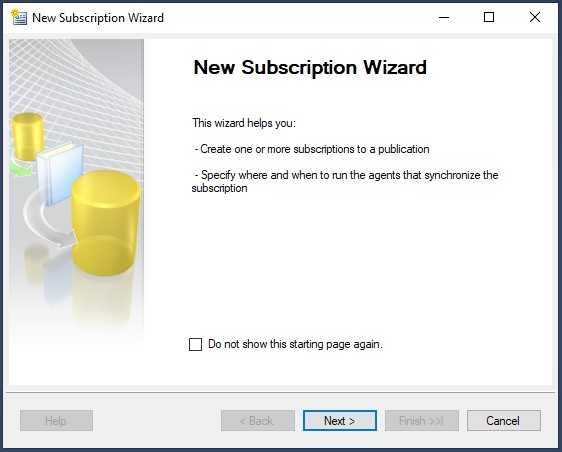


Hình 2.28. Kiểm tra kết quả View Snapshot Agent Status

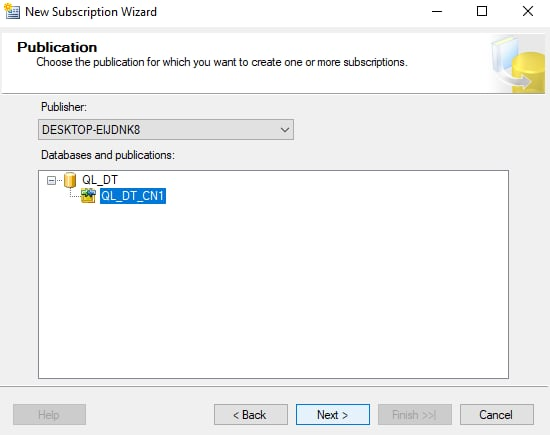
### Tạo Subscriptions



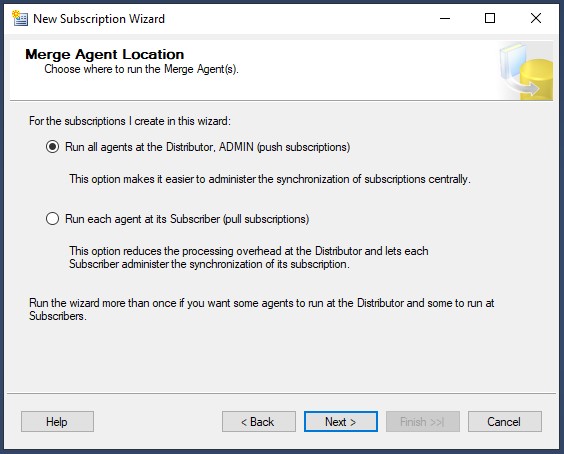
*Hình 2.29. Vào mục Local Publication trong mục QL\_DT\_CN1 chuột phải chọn New Subscriptions..*



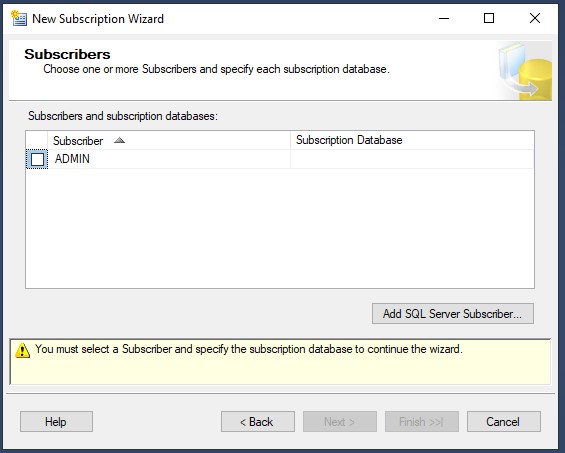
Hình 2.30. Chọn next



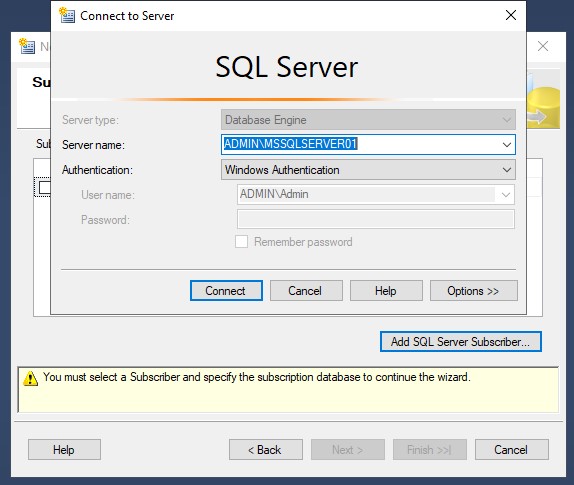
*Hình 2.31. Chọn QL\_DT\_CN1 và nhấn next*



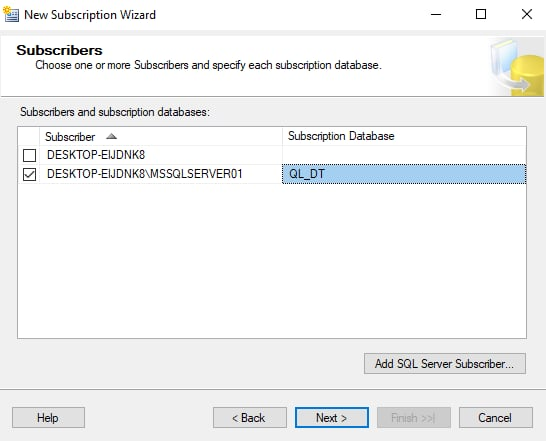
*Hình 2.32. Để mặc định và chọn next*



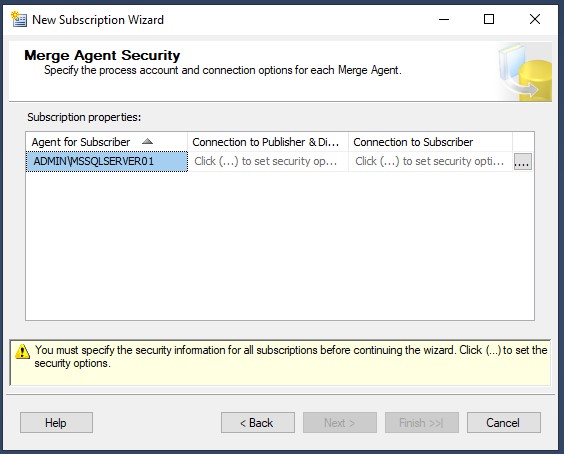
*Hình 2.33. Nhấn Add SQL Server Subscriber*



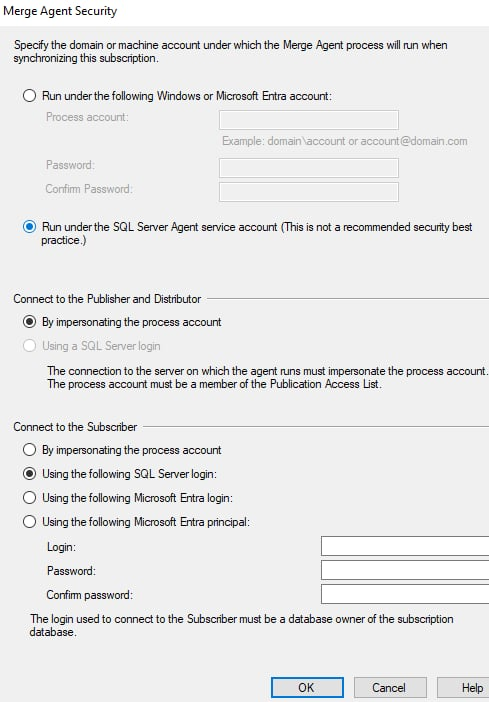
*Hình 2.34. Chọn SQL Server cần Add*



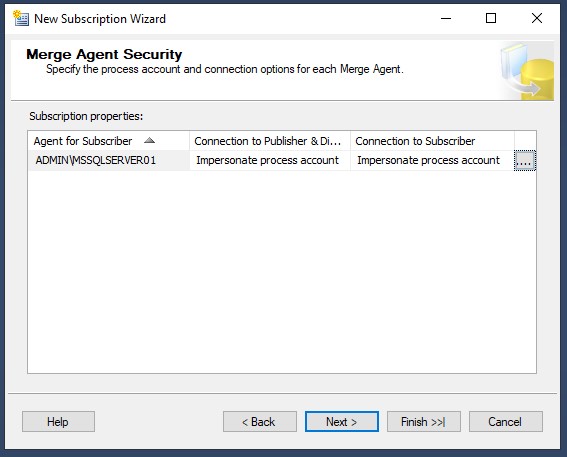
*Hình 2.35. Để mặc định và chọn next*



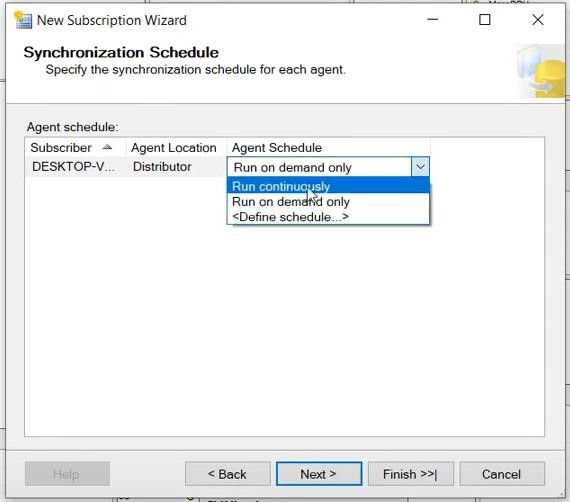
Hình 2.36. Click vào dấu …



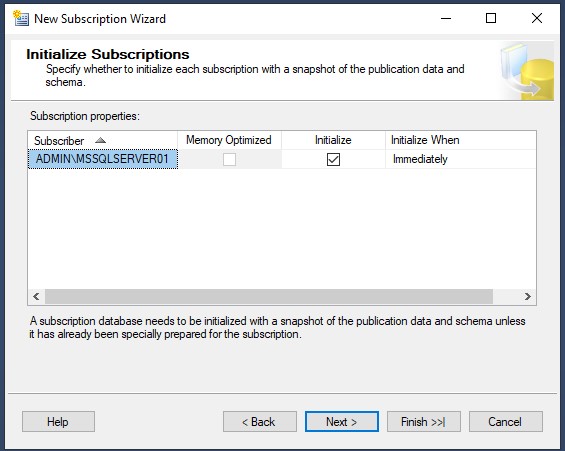
*Hình 2.37. Click chọn và nhấn OK*



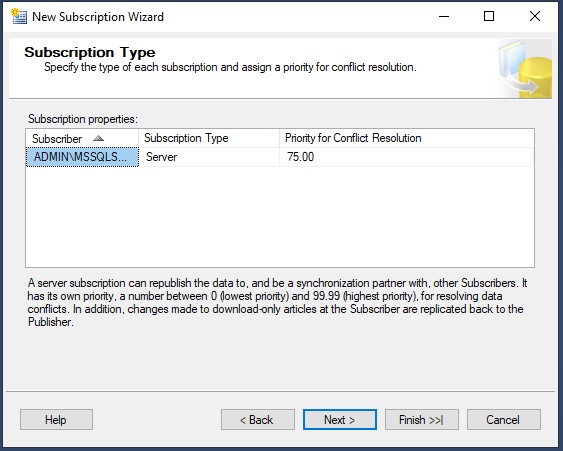
*Hình 2.38. Để mặc định và chọn next*



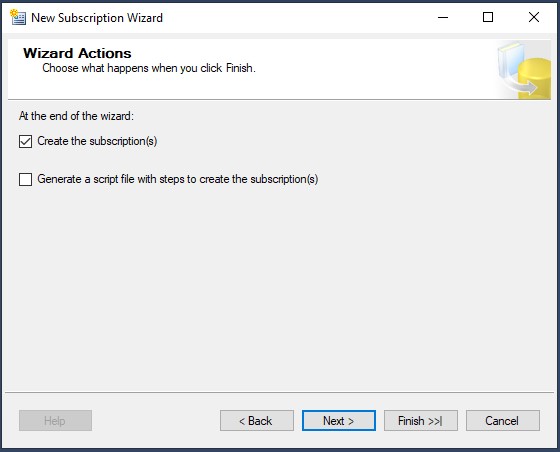
*Hình 2.39. Chọn Run continuously*



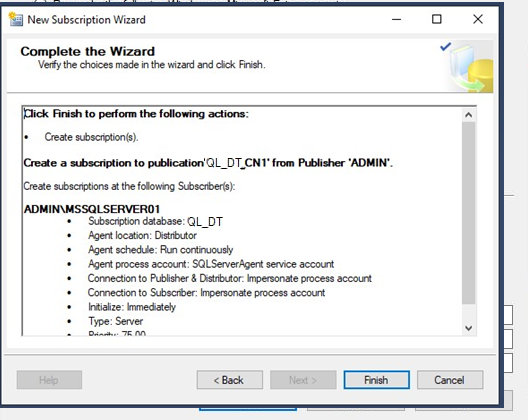
*Hình 2.40. Để mặc định và chọn next*



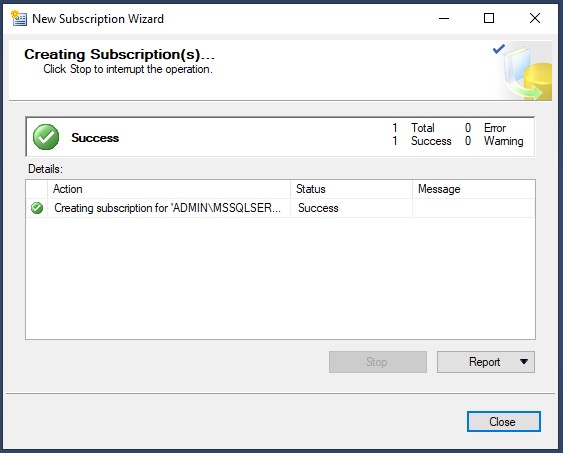
Hình 2.41. Để mặc định và chọn next



Hình 2.42. Để mặc định và chọn next



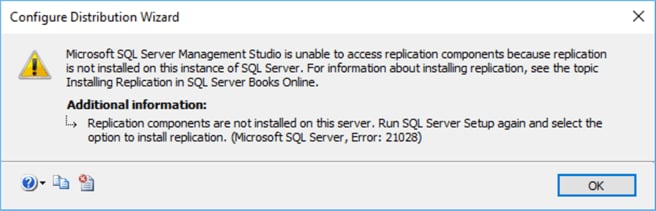
*Hình 2.43. Nhấn finish và chờ kết quả*



*Hình 2.44. Kết quả sau khi cài đặt*

## MỘT SỐ LỖI THƯỜNG GẶP

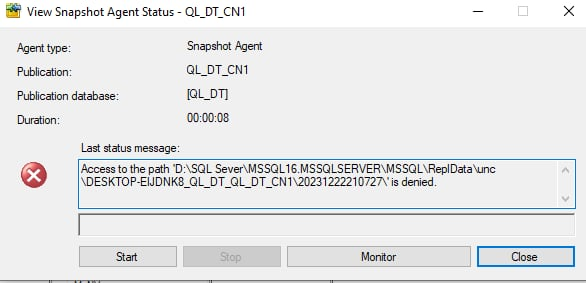
### Chưa cài đặt sql replication



Hình . Lỗi chưa cài đặt SQL replication

Giải pháp: cài đặt thêm SQL replication cho bản SQL hiện tại (ở SQL server installation center)

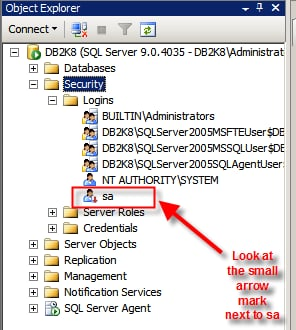
### Lỗi chưa sharing folder cho folder mà mình tạo ra phía trên



Hình. Lỗi chưa mở quyền đọc và xem của thư mục lưu các Database phân mảnh

Giải pháp: mở quyền đọc và xem cho thư mục đó.

### Không sử dụng tài khoản SA



Hình. Sử dụng tài khoản Sa

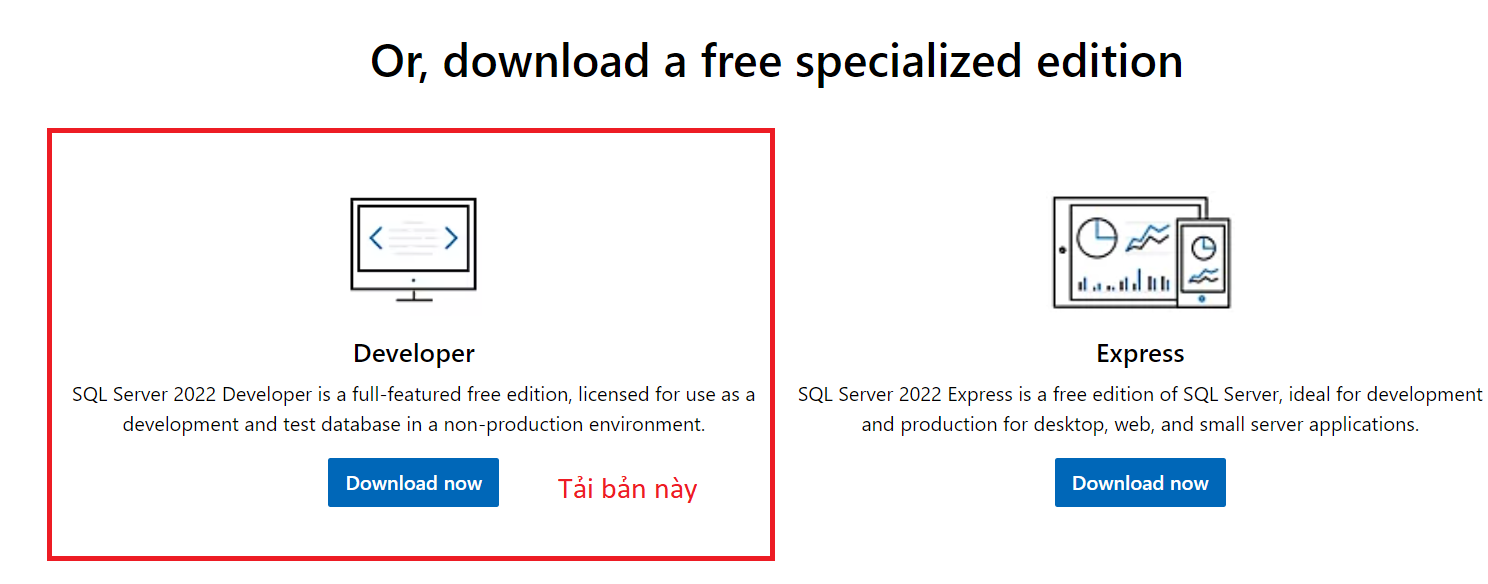
### Chưa khởi chạy SQL Server Agent

A screenshot of a computer error

Description automatically generated

Hình. Lỗi khi chưa khởi chạy SQL Server Agent

### Sử dụng bản SQL Server Express

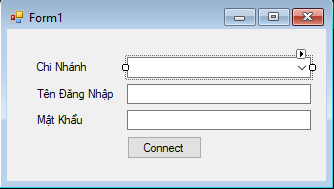


Hình. Đảm bảo cài đặt bản dành cho Developer để có thể kích hoạt SQL Agent

# THIẾT KẾ CHỨC NĂNG.

## ĐĂNG NHẬP

### Màn Hình Đăng Nhập.



Hình. Giao diện đăng nhập

### Thao Tác.

private void btnConnect\_Click(object sender, EventArgs e)

{

if (txtDangNhap.Text.Trim() == "")

{

MessageBox.Show("Tài khoản đăng nhập không được trống", "", MessageBoxButtons.OK);

return;

}

DBMain.mlogin = txtDangNhap.Text.Trim();

DBMain.password = txtMatKhau.Text;

if (DBMain.KetNoi() == 0)

return;

// 0 - trả về chi nhánh hiện tại

SqlDataReader myReader;

DBMain.mChinhanh = cmbChiNhanh.SelectedIndex;

//DBMain.bds\_dspm = ds;

DBMain.mloginDN = DBMain.mlogin;

DBMain.passwordDN = DBMain.password;

string strLenh = "exec KiemTraDangNhap '" + DBMain.mlogin + "'";

myReader = DBMain.ExecSqlDataReader(strLenh);

if (myReader == null) return;

myReader.Read();

DBMain.username = myReader.GetString(0); // Lay user name

if (Convert.IsDBNull(DBMain.username))

{

MessageBox.Show("User không đủ quyền truy cập ! Xin vui lòng xem lại cơ sở dữ liệu.", "", MessageBoxButtons.OK);

return;

}

DBMain.mGroup = myReader.GetString(2);

MessageBox.Show("Đăng nhập thành công !!!", "", MessageBoxButtons.OK);

if (txtDangNhap.Text == "QuanLy01" || txtDangNhap.Text == "QuanLy02")

{

frmDienThoai frm = new frmDienThoai();

frm.ShowDialog();

}

else

{

frmKhachHang frm = new frmKhachHang();

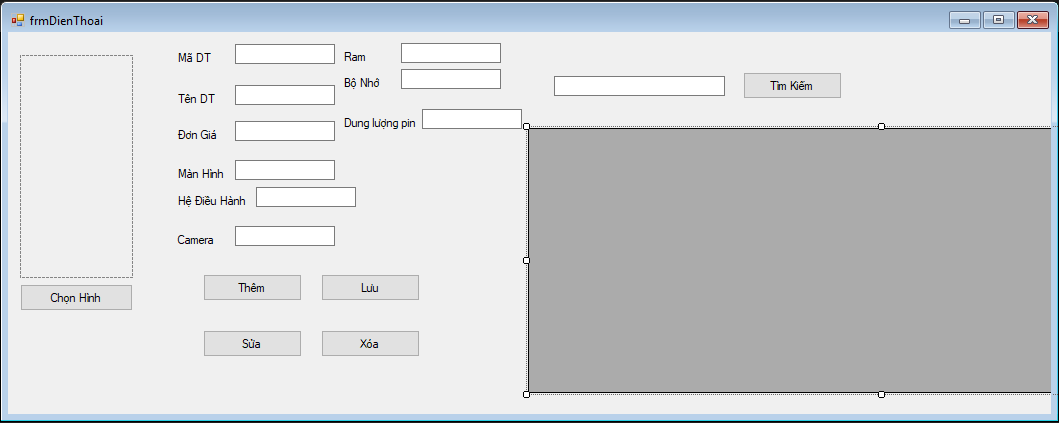
frm.ShowDialog();

}

}

## QUẢN LÍ ĐIỆN THOẠI

### Giao diện quản lí điện thoại



### Thao tác

* Load dữ liệu điện thoại.

private void frmDienThoai\_Load(object sender, EventArgs e)

{

HienThiDanhSachDT();

}

void HienThiDanhSachDT()

{

try

{

dt = db.LoadDienThoai();

dgvDIENTHOAI.RowTemplate.Height = 200;

dgvDIENTHOAI.AllowUserToAddRows = false;

dgvDIENTHOAI.DataSource = dt;

DataGridViewImageColumn imgCol = new DataGridViewImageColumn();

imgCol = (DataGridViewImageColumn)dgvDIENTHOAI.Columns[1];

imgCol.ImageLayout = DataGridViewImageCellLayout.Stretch;

}

catch (Exception)

{

MessageBox.Show("Không lấy được cơ sở dữ liệu trong bảng Điện Thoại!",

"Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

Chọn Hình.

private void btnChonHinh\_Click(object sender, EventArgs e)

{

OpenFileDialog dialog = new OpenFileDialog();

dialog.Filter = "png files(\*.png)|\*.png|jpg files(\*.jpg)|\*.jpg|All files(\*.\*)|\*.\*";

if(dialog.ShowDialog() == DialogResult.OK)

{

imgLocation = dialog.FileName.ToString();

pictureBox1.ImageLocation = imgLocation;

}

}

Thêm.

private void btnThem\_Click(object sender, EventArgs e)

{

them = true;

}

Lưu

private void btnLuu\_Click(object sender, EventArgs e)

{

byte[] images = null;

FileStream Streem = new FileStream(imgLocation, FileMode.Open, FileAccess.Read);

BinaryReader brs = new BinaryReader(Streem);

images = brs.ReadBytes((int)Streem.Length);

bool f = false;

if (them)

{

f = db.ThemDienThoai(ref err, this.txtMaDT.Text,images, this.txtTenDT.Text, int.Parse(this.txtDonGia.Text), this.txtManHinh.Text, this.txtHeDieuHanh.Text,this.txtCamera.Text,this.txtRam.Text,this.txtBoNho.Text,this.txtDungLuongPin.Text);

MessageBox.Show("Thêm Thành Công");

HienThiDanhSachDT();

}

else

{

f = db.SuaDienThoai(ref err, this.txtMaDT.Text, images, this.txtTenDT.Text, int.Parse(this.txtDonGia.Text), this.txtManHinh.Text, this.txtHeDieuHanh.Text, this.txtCamera.Text, this.txtRam.Text, this.txtBoNho.Text, this.txtDungLuongPin.Text);

MessageBox.Show("Sửa Thành Công");

HienThiDanhSachDT();

}

}

Xóa.

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

// Lấy thứ tự record hiện hành

int r = dgvDIENTHOAI.CurrentCell.RowIndex;

string MaDT =

dgvDIENTHOAI.Rows[r].Cells[0].Value.ToString();

DialogResult traloi;

// Hiện hộp thoại hỏi đáp

traloi = MessageBox.Show("Bạn chắc chắn xóa điện thoại này ?", "Trả lời",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// Kiểm tra có nhắp chọn nút Ok không?

string err = "";

if (traloi == DialogResult.Yes)

{

// Thực hiện câu lệnh SQL

bool f = db.XoaDienThoai(ref err, this.txtMaDT.Text);

if (f)

{

HienThiDanhSachDT();

MessageBox.Show("Đã xóa xong!");

}

else

{

MessageBox.Show("Không xóa được!\n\r" + "Lỗi:" + err);

}

}

else

{

// Thông báo

MessageBox.Show("Không thực hiện việc xóa mẫu tin!");

}

}

catch (SqlException)

{

MessageBox.Show("Không xóa được. Lỗi rồi!!!");

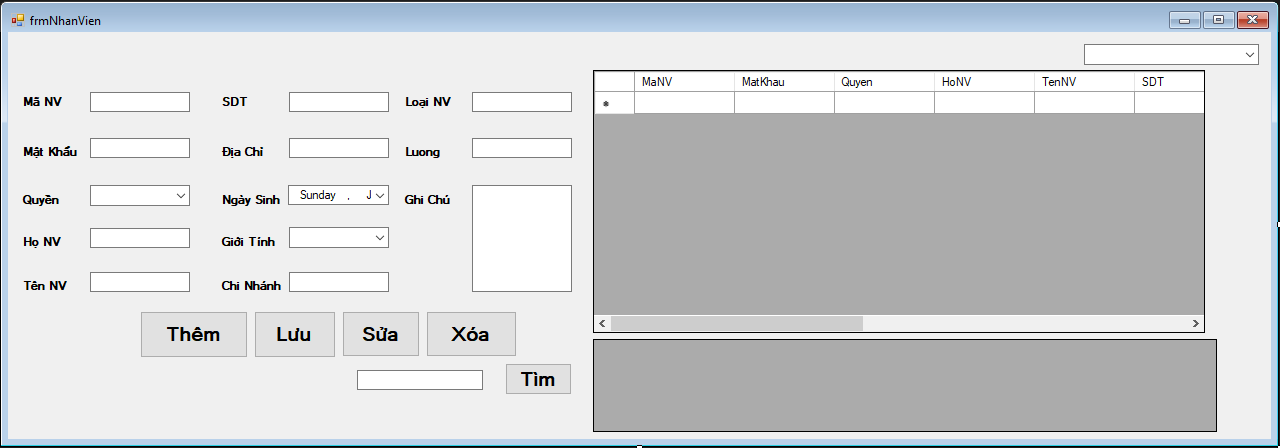
}

//Đóng kết nối

}

## QUẢN LÍ NHÂN VIÊN

### Giao diện quản lí nhân viên



### Thao tác

Load danh sách nhân viên

void LoadDanhSachNhanVien()

{

try

{

this.loadNhanVienTableAdapter.Connection.ConnectionString = DBMain.connstr;

this.loadNhanVienTableAdapter.Fill(this.xuLy.LoadNhanVien);

}

catch (Exception)

{

MessageBox.Show("Không lấy được cơ sở dữ liệu trong bảng Nhân Viên!",

"Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

Thêm nhân viên

private void btnThem\_Click(object sender, EventArgs e)

{

them = true;

}

Lưu nhân viên

private void btnLuu\_Click(object sender, EventArgs e)

{

try

{

bool f = false;

if (them)

{

f = db.ThemNhanVien(ref err, this.txtMaNV.Text, this.txtMatKhau.Text, this.cmbQuyen.Text, this.txtHoNV.Text, this.txtTenNV.Text, int.Parse(this.txtSDT.Text), this.txtDiaChi.Text, DateTime.Parse(this.dateTimePicker1.Text), this.cmbGioitinh.Text, this.txtChiNhanh.Text, this.txtLoaiNV.Text, int.Parse(this.txtLuong.Text), this.txtGhiChu.Text);

if (f == true)

{

MessageBox.Show(err);

this.loadNhanVienTableAdapter.Connection.ConnectionString = DBMain.connstr;

this.loadNhanVienTableAdapter.Fill(this.xuLy.LoadNhanVien);

}

else

{

MessageBox.Show(err);

}

}

else

{

f = db.SuaNhanVien(ref err, this.txtMaNV.Text, this.txtMatKhau.Text, this.cmbQuyen.Text, this.txtHoNV.Text, this.txtTenNV.Text, int.Parse(this.txtSDT.Text), this.txtDiaChi.Text, DateTime.Parse(this.dateTimePicker1.Text), this.cmbGioitinh.Text, this.txtChiNhanh.Text, this.txtLoaiNV.Text, int.Parse(this.txtLuong.Text), this.txtGhiChu.Text);

if (f == true)

{

MessageBox.Show("Sửa Nhân Viên Thành Công");

this.loadNhanVienTableAdapter.Connection.ConnectionString = DBMain.connstr;

this.loadNhanVienTableAdapter.Fill(this.xuLy.LoadNhanVien);

}

else

{

MessageBox.Show(err);

}

}

}

catch

{

}

}

Xóa nhân viên

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

// Lấy thứ tự record hiện hành

int r = dgvNHANVIEN.CurrentCell.RowIndex;

string MaNV =

dgvNHANVIEN.Rows[r].Cells[0].Value.ToString();

DialogResult traloi;

// Hiện hộp thoại hỏi đáp

traloi = MessageBox.Show("Bạn chắc chắn xóa nhân viên này ?", "Trả lời",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// Kiểm tra có nhắp chọn nút Ok không?

string err = "";

if (traloi == DialogResult.Yes)

{

// Thực hiện câu lệnh SQL

bool f = db.XoaNhanVien(ref err, this.txtMaNV.Text);

if (f)

{

// Cập nhật lại DataGridView

this.loadNhanVienTableAdapter.Connection.ConnectionString = DBMain.connstr;

this.loadNhanVienTableAdapter.Fill(this.xuLy.LoadNhanVien);

// Thông báo

MessageBox.Show("Đã xóa xong!");

}

else

{

MessageBox.Show("Không xóa được!\n\r" + "Lỗi:" + err);

}

}

else

{

// Thông báo

MessageBox.Show("Không thực hiện việc xóa mẫu tin!");

}

}

catch (SqlException)

{

MessageBox.Show("Không xóa được. Lỗi rồi!!!");

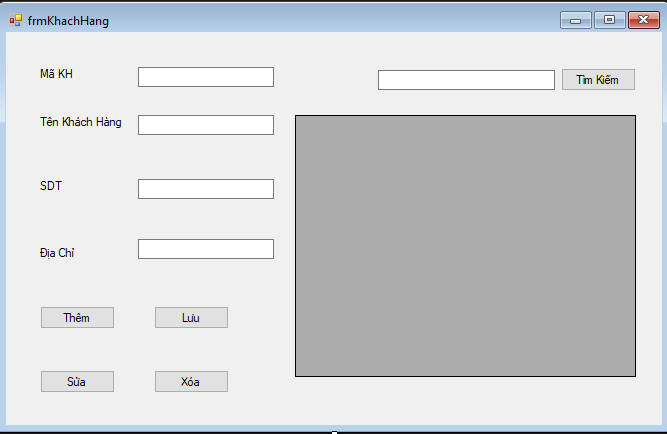
}

//Đóng kết nối

}

## GIAO DIỆN QUẢN LÝ KHÁCH HÀNG

### Màn hình quản lí khách hàng



**G**

### Thao tác

Load danh sách khách hàng.

private void frmKhachHang\_Load(object sender, EventArgs e)

{

HienThiDanhSachKH();

}

void HienThiDanhSachKH()

{

try

{

kh = db.LoadKhachHang();

dgvKHACHHANG.DataSource = kh;

dgvKHACHHANG.AutoResizeColumns();

}

catch (Exception)

{

MessageBox.Show("Không lấy được cơ sở dữ liệu trong bảng Khách Hàng!",

"Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

Thêm khách hàng

private void btnThem\_Click(object sender, EventArgs e)

{

them = true;

}

Sữa khách hàng

private void btnSua\_Click(object sender, EventArgs e)

{

them = false;

}

Lưu

private void btnLuu\_Click(object sender, EventArgs e)

{

try

{

bool f = false;

if (them)

{

f = db.ThemKhachHang(ref err, this.txtMaKH.Text, this.txtTenKhachHang.Text, this.txtDiaChi.Text, int.Parse(this.txtSDT.Text));

if (f == true)

{

HienThiDanhSachKH();

}

else

{

MessageBox.Show(err);

}

}

else

{

f = db.SuaKhachHang(ref err, this.txtMaKH.Text, this.txtTenKhachHang.Text, this.txtDiaChi.Text, int.Parse(this.txtSDT.Text));

if (f == true)

{

HienThiDanhSachKH();

}

else

{

MessageBox.Show(err);

}

}

}

catch

{

}

}

Xóa

private void btnXoa\_Click(object sender, EventArgs e)

{

try

{

// Lấy thứ tự record hiện hành

int r = dgvKHACHHANG.CurrentCell.RowIndex;

string MaKH =

dgvKHACHHANG.Rows[r].Cells[0].Value.ToString();

DialogResult traloi;

// Hiện hộp thoại hỏi đáp

traloi = MessageBox.Show("Bạn chắc chắn xóa nhân viên này ?", "Trả lời",

MessageBoxButtons.YesNo, MessageBoxIcon.Question);

// Kiểm tra có nhắp chọn nút Ok không?

string err = "";

if (traloi == DialogResult.Yes)

{

// Thực hiện câu lệnh SQL

bool f = db.XoaKhachHang(ref err, this.txtMaKH.Text);

if (f)

{

// Cập nhật lại DataGridView

HienThiDanhSachKH();

// Thông báo

MessageBox.Show("Đã xóa xong!");

}

else

{

MessageBox.Show("Không xóa được!\n\r" + "Lỗi:" + err);

}

}

else

{

// Thông báo

MessageBox.Show("Không thực hiện việc xóa mẫu tin!");

}

}

catch (SqlException)

{

MessageBox.Show("Không xóa được. Lỗi rồi!!!");

}

//Đóng kết nối