**TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG**



**KHOA THỐNG KÊ – TIN HỌC**

**- - - 🙞 🕮** **🙜 - - -**



**BÁO CÁO DỰ ÁN**

**QUẢN LÝ ỨNG DỤNG NHẬP HÀNG CỬA HÀNG ĐIỆN MÁY BÌNH KÍNH**

**MÔN: QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Giáo viên hướng dẫn:** Th.S Cao Thị Nhâm

**Nhóm : 3**

**Sinh viên thực hiện:** Phạm Vương Phương Anh

Bùi Thị Thúy

Lê Xuân Nhật

Trần Văn Hùng

**Lớp:** 45K21.2

***Đà Nẵng, 11/2021***

**PHẦN TRĂM ĐÓNG GÓP**

|  |  |
| --- | --- |
| Họ và tên | Phần trăm công việc |
| Phạm Vương Phương Anh | 25% |
| Trần Văn Hùng | 25% |
| Lê Xuân Nhật | 25% |
| Bùi Thi Thúy | 25% |

**LỜI CẢM ƠN**

Để hoàn thành đề tài nghiên cứu này, nhóm chúng em xin được gửi lời cảm ơn sâu sắc tới cô Cao Thị Nhâm đã tận tình hướng dẫn, chỉ dạy trong suốt quá trình nhóm thực hiện đề tài.

Chúng em xin được cảm ơn cô Cao Thị Nhâm đã hỗ trợ giảng dạy và cung cấp tài liệu trong thời gian vừa qua.

Trong thời gian qua, mặc dù đã rất cố gắng nhưng do một phần kiến thức còn hạn chế nên bài báo cáo khó tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý từ cô để bài báo cáo trở nên hoàn thiện hơn.

Trân trọng cảm ơn !

# MỤC LỤC

[MỤC LỤC 4](#_Toc88852175)

[DANH MỤC HÌNH ẢNH 6](#_Toc88852176)

[I. Hóa đơn, chứng từ 8](#_Toc88852177)

[II. Thiết kế cơ sở dữ liệu: 11](#_Toc88852178)

[1. Thiết kế cơ sở dữ liệu mức khái niệm 11](#_Toc88852179)

[1.1. Xây dựng mô hình E-R cho hóa đơn bán hàngChọn lọc thông tin: 11](#_Toc88852180)

[1.2. Các thực thể 12](#_Toc88852181)

[1.3. Các quan hệ 12](#_Toc88852182)

[2. Thiết kế cơ sở dữ liệu mức logic 13](#_Toc88852183)

[III. Xây dựng cơ sở dữ liệu: 15](#_Toc88852184)

[IV. Xác định và tạo index cho các thuộc tính: 18](#_Toc88852185)

[V. Xây dựng cơ chế bảo mật tương ứng cho cơ sở dữ liệu vừa tạo : 20](#_Toc88852186)

[1. Authentication 20](#_Toc88852187)

[2. Authority 21](#_Toc88852188)

[VI.   Xây dựng cơ chế backup dữ liệu: 25](#_Toc88852189)

[1. Thực hiện backup thủ công 25](#_Toc88852190)

[2. Thực hiện backup tự động: 26](#_Toc88852191)

[VII.  Phát triển ứng dụng: 30](#_Toc88852192)

[1. Form Đăng nhập 30](#_Toc88852193)

[2. Form Quản lý chung 32](#_Toc88852194)

[3. Form Hàng 34](#_Toc88852195)

[4. Form Nhà cung cấp 39](#_Toc88852196)

[5. Form thống kê 44](#_Toc88852197)

[6. Form Hóa đơn 45](#_Toc88852198)

[VIII. Phương án giải quyết vấn đề dữ liệu lớn: 53](#_Toc88852199)

[IX.   Đề phòng phương án bị tấn công bằng SQL Injection: 58](#_Toc88852200)

[TÀI LIỆU THAM KHẢO 59](#_Toc88852201)

# DANH MỤC HÌNH ẢNH

[Hình 1 Hóa đơn giá trị gia tăng 7](#_Toc88851867)

[Hình 2 Hóa đơn giá trị gia tăng 8](#_Toc88851868)

[Hình 3 Hóa đơn giá trị gia tăng 9](#_Toc88851869)

[Hình 4 Thực thể liên kết ERD 12](#_Toc88851870)

[Hình 5 sơ đồ quan hệ 13](#_Toc88851871)

[Hình 6 Tạo thành công index 18](#_Toc88851872)

[Hình 7 Tạo tài khoản cho Quản lý 19](#_Toc88851873)

[Hình 8 Tạo tài khoản cho nhân viên 20](#_Toc88851874)

[Hình 9 Phân quyền membership cho nhân viên 21](#_Toc88851875)

[Hình 10 Phân quyền cho tài khoản nhân viên 22](#_Toc88851876)

[Hình 10 Phân quyền cho tài khoản quản lý 22](#_Toc88851877)

[Hình 11 Phân quyền cho tài khoản quản lý 23](#_Toc88851878)

[Hình 11 Backup thủ công 24](#_Toc88851879)

[Hình 11 File Backup 25](#_Toc88851880)

[Hình 12 Back Up Database Task 25](#_Toc88851881)

[Hình 13 Thiết lập trong New Job Step 26](#_Toc88851882)

[Hình 14 Thiết lập Job Shedule - Backup 27](#_Toc88851883)

[Hình 15 Job Properties - Backup 28](#_Toc88851884)

[Hình 16 From đăng nhập 29](#_Toc88851885)

[Hình 16 Form quản lý chung 31](#_Toc88851886)

[Hình 17 Form Hàng hóa 33](#_Toc88851887)

[Hình 18 Form hóa đơn 38](#_Toc88851888)

[Hình 19 Form nhà cung cấp 46](#_Toc88851889)

[Hình 20 Form thống kê 51](#_Toc88851890)

[Hình 20 Tạo tài khoản AWS 53](#_Toc88851891)

[Hình 21 Tạo cơ sở dữ liệu 53](#_Toc88851892)

[Hình 22 Vào bảng điều khiển 54](#_Toc88851893)

[Hình 23 Tạo thành công 55](#_Toc88851894)

[Hình 24 Kết nối Microsoft SQL Server 56](#_Toc88851895)

[Hình 25 Xóa phiên cơ sở dữ liệu 56](#_Toc88851896)

# Hóa đơn, chứng từ

**Table

Description automatically generated**

Hình 1 Hóa đơn giá trị gia tăng

**Table

Description automatically generated**

Hình 2 Hóa đơn giá trị gia tăng

**Table

Description automatically generated**

Hình 3 Hóa đơn giá trị gia tăng

# Thiết kế cơ sở dữ liệu:

## Thiết kế cơ sở dữ liệu mức khái niệm

### Xây dựng mô hình E-R cho hóa đơn bán hàngChọn lọc thông tin:

|  |  |  |
| --- | --- | --- |
| **Từ trong HSDL** | **Từ rõ nghĩa** | **Viết tắt** |
| Mẫu số | Mẫu số hóa đơn | MauSoHD |
| Ký hiệu | Ký hiệu hóa đơn | KyHieuHD |
| Số | Mã hóa đơn | MaHD |
| Ngày | Ngày lập hóa đơn | NgayLapHD |
|  | | |
| Đơn vị bán hàng | Tên đơn vị bán hàng | TenNCC |
| Mã số thuế | Mã số thuế người bán | MaNCC |
| Địa chỉ | Địa chỉ người bán | DiaChi |
| Điện thoại | SDT người bán | Sdt |
| Thư điện tử | Thư điện tử người bán | TdtNCC |
| Số TK | STK của người bán | SoTKNCC |
| Ngân hàng | Ngân hàng mà stk người bán dùng | NganHangNCC |
|  | | |
| Họ tên người mua hàng | Họ tên người mua hàng | TenNV |
| Tên đơn vị | Tên đơn vị mua hàng | TenDVNM |
| Mã số thuế | Mã số thuế người mua | MaNV |
| Địa chỉ | Địa chỉ người mua | DiaChiNM |
| Hình thức thanh toán | Hình thức thanh toán | HinhThucTT |
| Ngân hàng | Ngân hàng dùng để giao dịch | NganHangNM |
|  | | |
| Tên hàng hóa/dịch vụ | Tên hàng hóa/dịch vụ | TenHH |
| Đơn vị tính | Đơn vị tính | DonVi |
| Số lượng | Số lượng | SoLuong |
| Đơn giá | Đơn giá | DonGia |
| Thành tiền | Thành tiền | ThanhTien |
| Cộng tiền hàng | Cộng tiền hàng | CongTien |
| Thuế suất thuế GTGT | Thuế suất thuế GTGT | VATrate |
| Tiền thuế GTGT | Tiền thuế GTGT | TienVAT |
| Tổng tiền thanh toán | Tổng tiền thanh toán | TongTT |
| Số tiền viết bằng chữ | Số tiền viết bằng chữ | TongTTchu |

### 1.2. Các thực thể

**Xác định thực thể, thuộc tính**

NHACUNGCAP(MaNCC, TenNCC, DiaChi, Sdt)

NHANVIEN(MaNV, TenNV, DiaChiNM, SDTNM)

HANGHOA(MaHH, TenHH, DonVi, DonGia)

### 1.3. Các quan hệ

**Xác định quan hệ**

NHẬP: Bao gồm các thuộc tính còn lại

**Vẽ sơ đồ thực thể liên kết ERD**

|  |
| --- |
|  |

Hình 4 Thực thể liên kết ERD

## Thiết kế cơ sở dữ liệu mức logic

**B1: Chuyển thực thể, thu được:**

NHACUNGCAP(MaNCC, TenNCC, DiaChi, Sdt)

HANGHOA(MaHH, TenHH, DonVi, DonGia)

**B2: Chuyển quan hệ, thu được:**

NHẬP (MaHDN, MaHH, MaNV, HinhThucTT, NganHangNM, SoLuongNhap, NgayNhap, ThanhTien, TongTien, TienVAT)

**B3: Chuẩn hóa**

NHẬP (MaHDN, MaCC, NgayNhap, TongTien)

NHẬP\_CHI TIẾT (MaHDN, MaHH, SoLuongNhap, ThanhTien)

**B4: Vẽ sơ đồ quan hệ**

Diagram

Description automatically generated

Hình 5 sơ đồ quan hệ

# Xây dựng cơ sở dữ liệu:

* Tạo database :

USE master;

GO

CREATE DATABASE nhaphang

go

use nhaphang

go

Bả

* Tạo bảng NHACUNGCAP

--Tạo bảng NHACUNGCAP

Create table NHACUNGCAP

(

MaNCC char(6),

TenNCC Nvarchar(50),

DiaChi Nvarchar(100),

Sdt varchar(15),

primary key(MaNCC)

)

Go

Bảng NHACUNGCAP

Graphical user interface, application, table

Description automatically generated

* Tạo bảng HANGHOA

Create table HANGHOA

(

MaHH varchar(10),

TenHH Nvarchar(100),

DonVi Nvarchar(10),

DonGia numeric(15,0),

primary key(MaHH)

)

Go

Bảng HANGHOA

Graphical user interface, application, table

Description automatically generated

* Tạo bảng NHAP

Create table NHAP

(

MaHDN varchar(10),

MaNCC char(6),

NgayLapHD date,

TongTien varchar(20),

primary key(MaHDN),

foreign key (MaNCC) references NHACUNGCAP

)

Go

Bảng NHAP

Table

Description automatically generated

* Tạo bảng NHAP\_CHITIET

Create table NHAP\_\_CHITIET

(

MaHDN varchar(10),

MaHH varchar(10),

SoLuong int,

ThanhTien numeric(15,0),

primary key(MaHDN),

foreign key (MaHDN) references NHAP,

foreign key (MaHH) references HANGHOA

)

Go

Bảng NHAP\_CHITIET

Table

Description automatically generated

# Xác định và tạo index cho các thuộc tính:

Xác định và tạo các index cho các thuộc tính cần thiết. (Hãy giải thích vì sao lại đưa ra lựa chọn như vậy)

* Tạo index cho TenHH để khi tìm tên hàng hóa trong bảng HANGHOA nhanh hơn:

CREATE NONCLUSTERED INDEX IDX\_TenHH\_HANGHOA on HANGHOA(TenHH)

Khi tìm kiếm một hàng hóa mà thông qua tên hàng hóa khi sử dụng index này thì quá trình tìm kiếm sẽ được đẩy nhanh bởi vì các tên hàng hóa đã được sắp xếp theo thứ tự bảng chữ cái nên câu truy vấn sẽ không phải tìm kiếm cả một bảng để tìm kiếm tên hàng hóa đó

* Tạo index cho TenNCC để khi tim tên nhà cung cấp trong bảng NHACUNGCAP nhanh hơn:

CREATE NONCLUSTERED INDEX IDX\_TenNCC\_NHACUNGCAP ON NHACUNGCAP(TenNCC)

* Tạo index cho NgayLHD để khi tìm số hóa đơn trong ngày trong bảng NHAP nhanh hơn:

CREATE NONCLUSTERED INDEX IDX\_NgayLHD\_NHAP on NHAP(NgayLapHD)

Lý do chọn các dòng index trên:

Khi sử dụng các dòng index trên trong quá trình tìm kiếm sẽ được tiến hành nhanh chóng vì các thuộc tính ấy đã được sắp xếp theo thứ nên câu truy vấn sẽ không phải tìm kiếm cả bảng để tìm kiếm một bản ghi nào đó.

Ví dụ: Tìm bản ghi trong bảng NHACUNGCAP thông qua thuộc tính tên nhà cung cấp với mệnh đề WHERE

Select \* from NHACUNGCAP where TenNCC = N'Công ty Hằng Hữu'

SQL Server sẽ kích hoạt dòng index đã được tạo ở cột TenNCC. Khi đó tên của các nhà cung cấp sẽ được sắp xếp theo bảng chữ cái mà câu truy vấn sẽ tìm lần lượt các bản ghi lúc đó sẽ nhanh hơn so với việc phải tìm tất cả các bảng ghi trong bảng

Graphical user interface

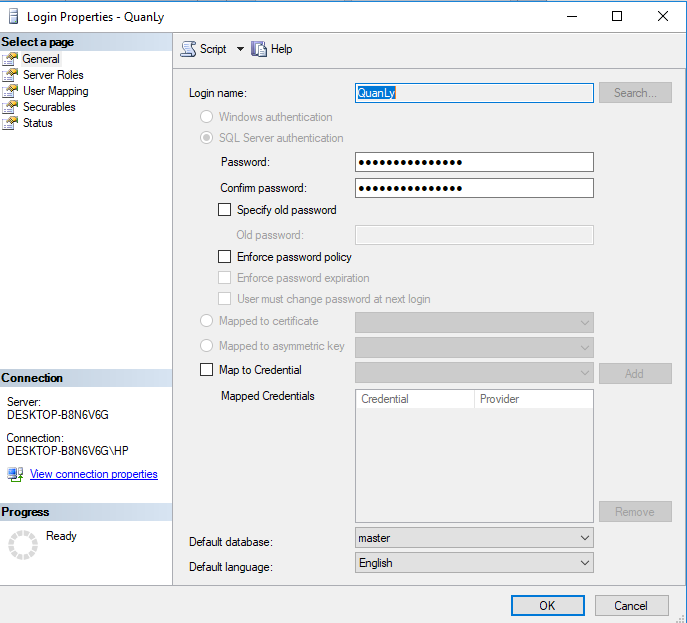
Description automatically generated

Hình 6 Tạo thành công index

# Xây dựng cơ chế bảo mật tương ứng cho cơ sở dữ liệu vừa tạo :

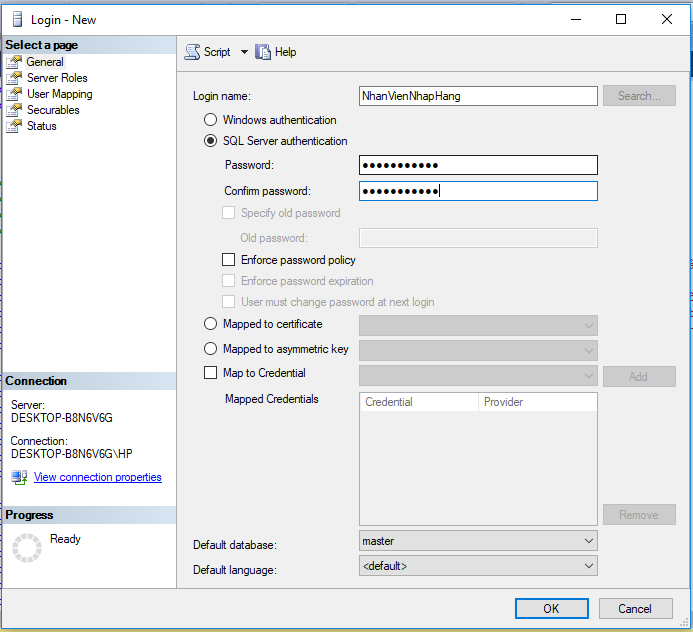
## Authentication

Tạo tài khoản QuanLy cho database nhaphang

****

Hình 7 Tạo tài khoản cho Quản lý

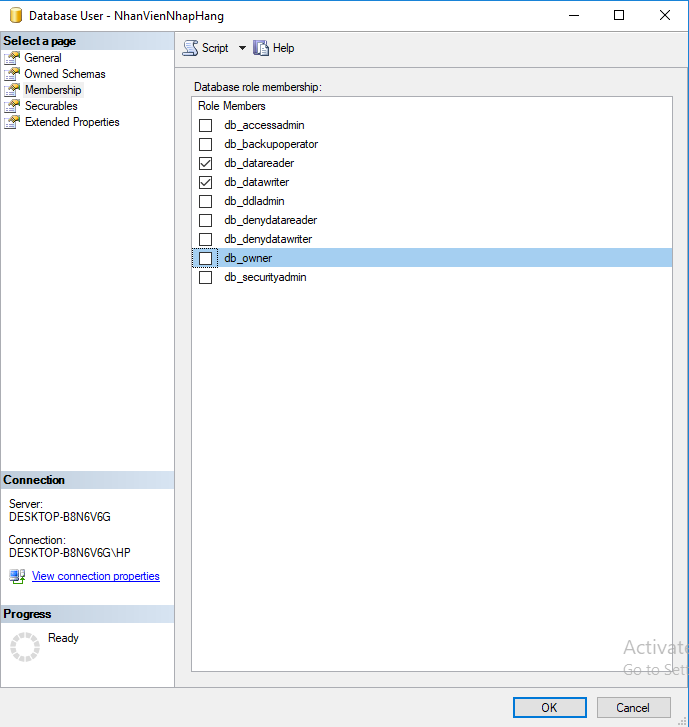
Tạo tài khoản NhanVienNhapHang cho database nhaphang

****

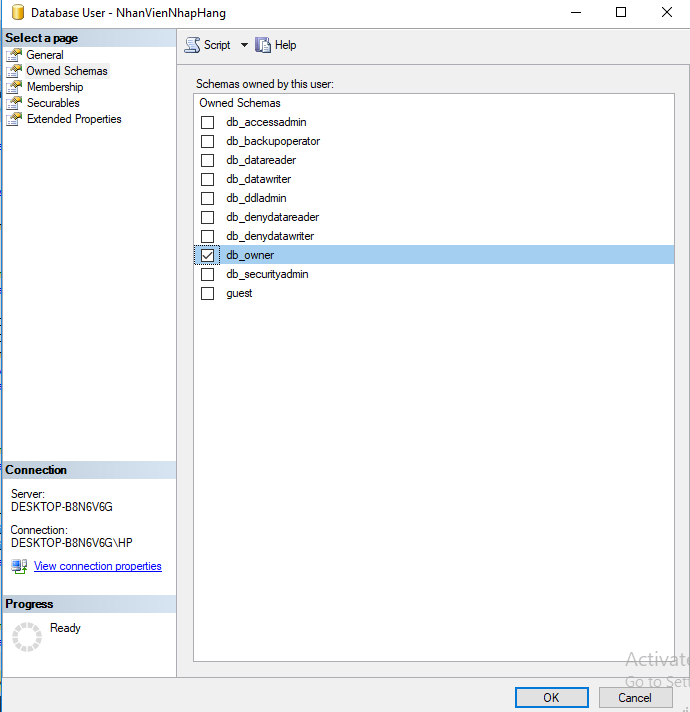
Hình 8 Tạo tài khoản cho nhân viên

## Authority

* Phân quyền cho nhân viên chỉ được xem

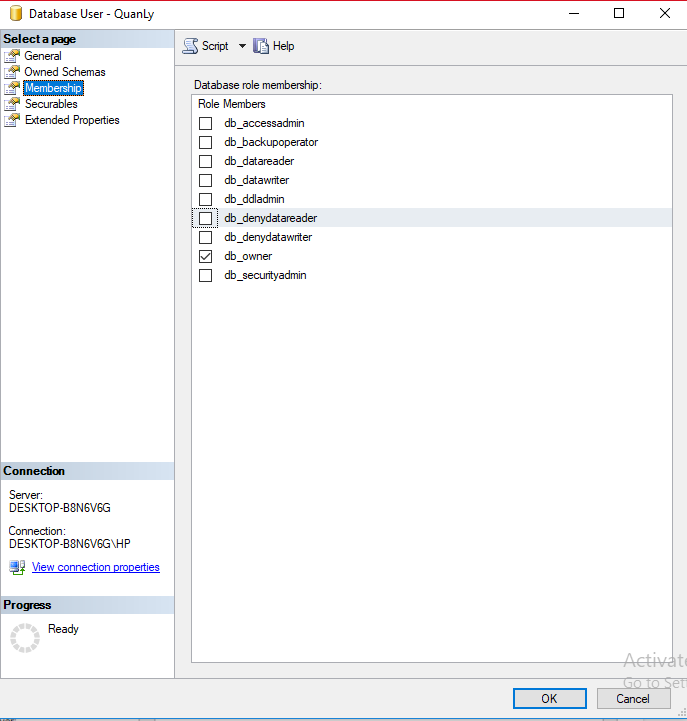


Hình 9 Phân quyền membership cho nhân viên

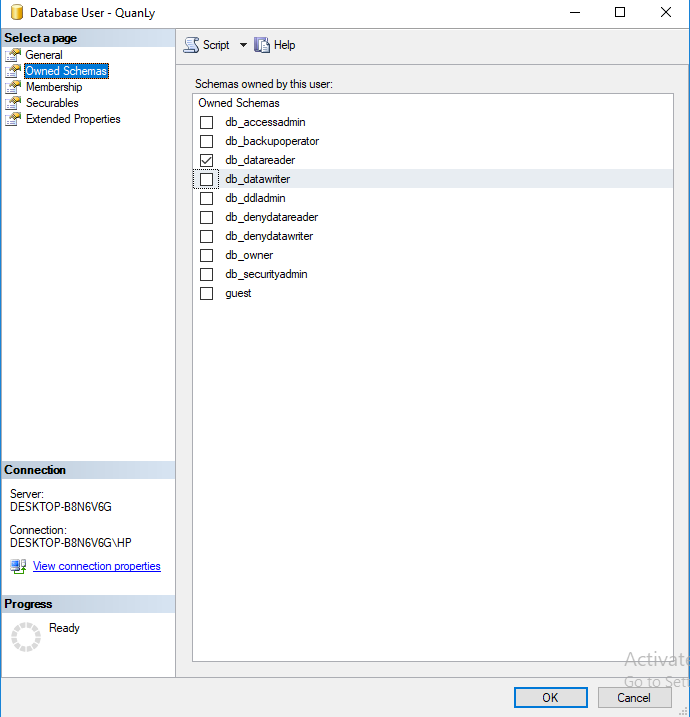
****

Hình 10 Phân quyền cho tài khoản nhân viên

* Phân quyền quản lý có mọi quyền sử dụng trên database nhaphang

****

Hình 10 Phân quyền cho tài khoản quản lý

****

Hình 11 Phân quyền cho tài khoản quản lý

# VI.   Xây dựng cơ chế backup dữ liệu:

## 1. Thực hiện backup thủ công

Graphical user interface, text, application, email

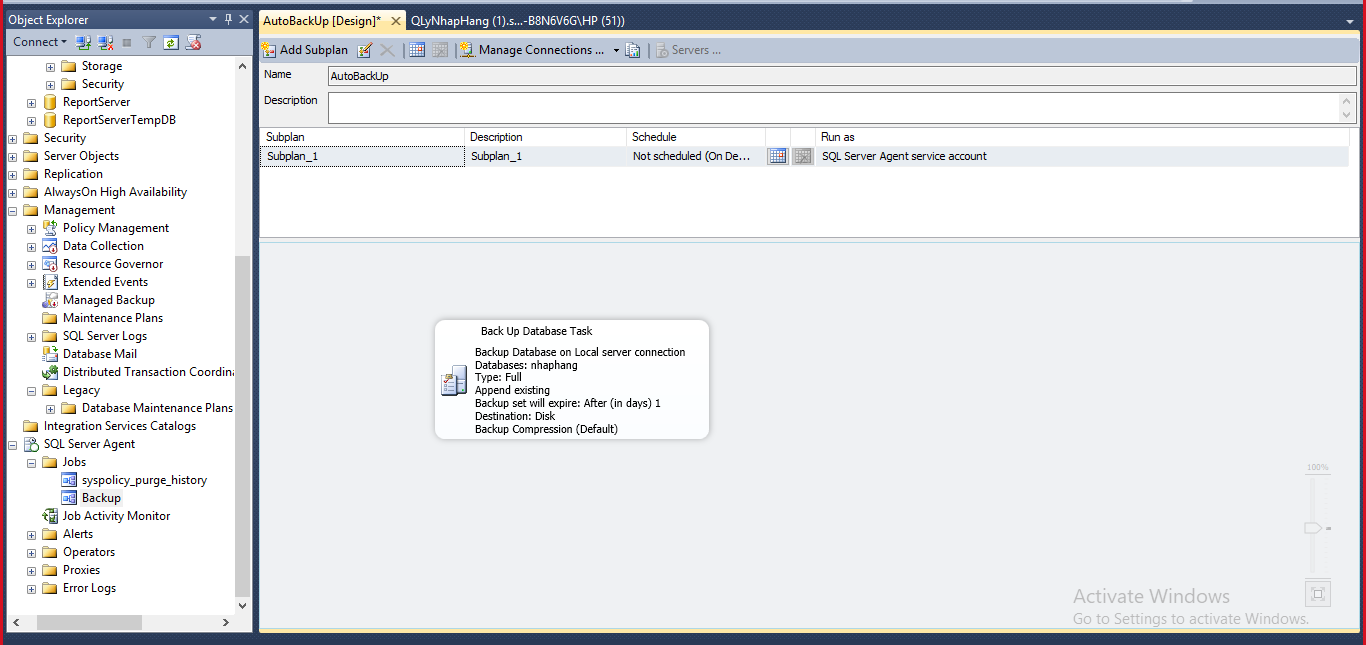
Description automatically generated

Hình 11 Backup thủ công

Graphical user interface, text, application, email

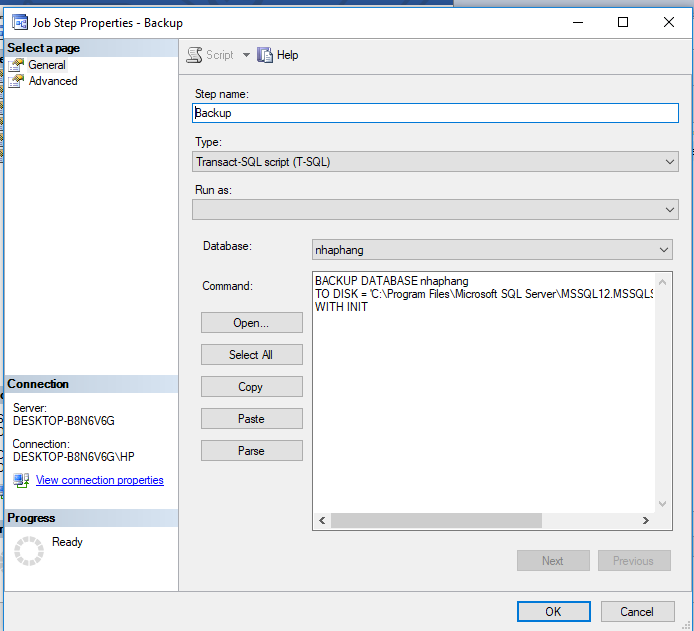
Description automatically generated

Hình 11 File Backup



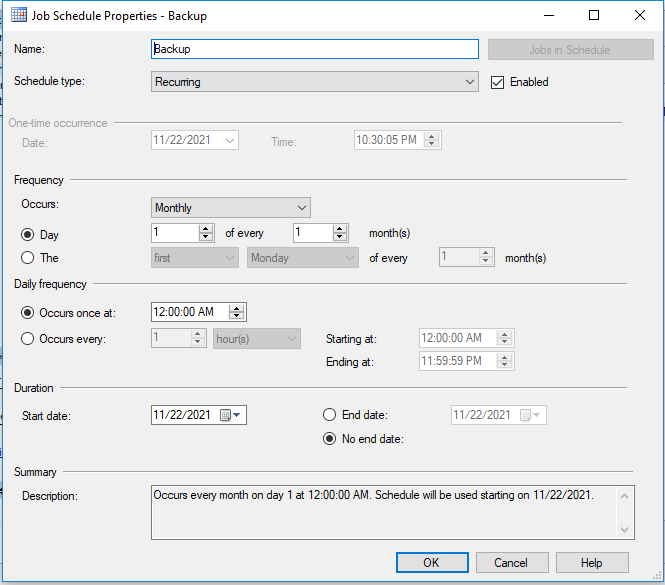
Hình 12 Back Up Database Task

## 2. Thực hiện backup tự động:



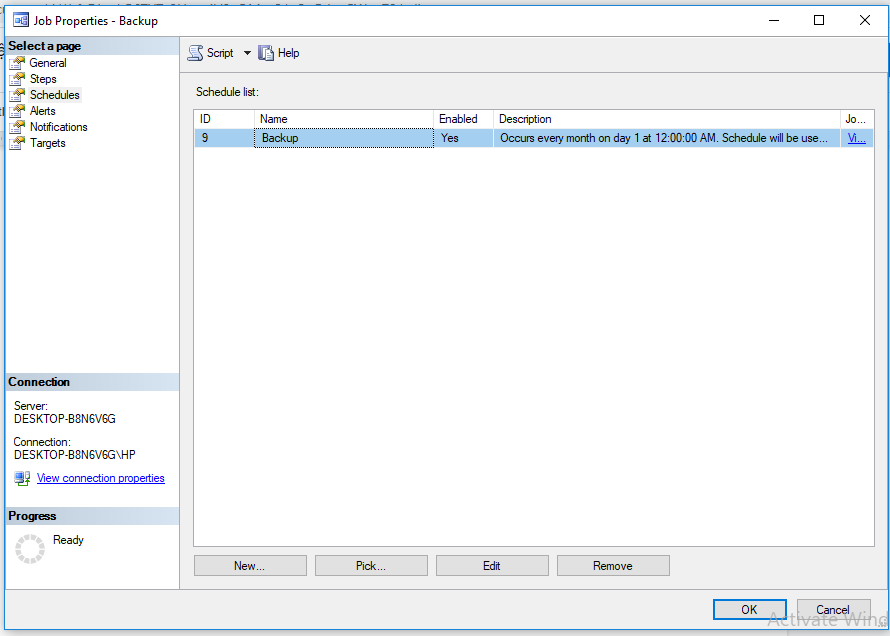
Hình 13 Thiết lập trong New Job Step

Ở cuối câu lệnh là mệnh đề “WITH INIT” để đảm bảo hàng ngày chạy nó sẽ ghi đè lên file hiện tại. Nếu không có mệnh đề này, các bản backup sẽ được ghi nối tiếp nhau trong cùng một file và file backup sẽ tăng kích thước lên nhanh chóng.



Hình 14 Thiết lập Job Shedule - Backup

Trong ô “Name” nhập vào “Backup”; ở ô “Occurs” chọn “Monthly” và Occurs at chọn “12:00:00 AM” vì như vậy backup job sẽ chạy hàng ngày vào lúc 0 giờ sáng. Lý do của việc chọn giờ như vậy là vì thường ban đêm database có ít hoạt động và do đó không ảnh hưởng nhiều đến user.



Hình 15 Job Properties - Backup

# VII.  Phát triển ứng dụng:

## Form Đăng nhập



Hình 16 From đăng nhập

- Các thành phần của form:

Textbox: Tên đăng nhập, mật khẩu

Button: Đăng nhập, thoát

- Chức năng:

+ Nút Đăng nhập: Kiểm tra thông tin tài khoản và mật khẩu. Nếu tài khoản và mật khẩu đúng, sẽ nhận được thông báo:” Đăng nhập thành công.” để bắt đầu giao diện. Ngược lại, thông báo “Đăng nhập thất bại”.

+ Nút Thoát: Thoát khỏi form đăng nhập.

* **Các sự kiện trong form**

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

{

SqlConnection conn = new SqlConnection("Data Source=.\\sqlExpress;Initial Catalog=Dangnhap;Integrated Security=True");

try

{

conn.Open();

string tk = txtTendangnhap.Text;

string mk = txtMatkhau.Text;

string sql = "select \* from taikhoan where Taikhoan='" + tk + "' and Matkhau='" + mk + "'";

SqlCommand cmd = new SqlCommand(sql, conn);

SqlDataReader dta = cmd.ExecuteReader();

if(dta.Read() == true)

{

MessageBox.Show("Đăng nhập thành công");

frmquanlichung KT = new frmquanlichung();

KT.Show();

this.Hide();

}

else

{

MessageBox.Show("Đăng nhập thất bại");

}

}

catch(Exception)

{

MessageBox.Show("Lỗi kết nối");

}

}

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

{

Application.Exit();

}

## Form Quản lý chung

Graphical user interface, text, application, Word

Description automatically generated

Hình 16 Form quản lý chung

Form quản lí chung có chức năng:

* Click vào **Quản lí hàng hóa** : Hiển thị sang form hàng
* Click vào **Quản lí nhà cung cấp:** Hiển thị sang form nhà cung cấp
* Click vào **Quản lí hóa đơn** : Hiển thị sang form hóa đơn
* Click vào **Thống kê hóa đơn nhập**: Hiển thị sang form thống kê

**Các sự kiện trong form**

public partial class frmquanlichung : Form

{

public frmquanlichung()

{

InitializeComponent();

}

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

{

frmhoadon hoadon = new frmhoadon();

hoadon.MdiParent = this;

hoadon.Show();

}

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

{

frmnhacungcap nhacungcap = new frmnhacungcap();

nhacungcap.MdiParent = this;

nhacungcap.Show();

}

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

{

frmhang hang = new frmhang();

hang.MdiParent = this;

hang.Show();

}

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

{

frmthongke thongke = new frmthongke();

thongke.MdiParent = this;

thongke.Show();

}

private void frmquanlichung\_FormClosing(object sender, FormClosingEventArgs e)

{

if (MessageBox.Show("Bạn có muốn thoát chương trình?", "Thông báo", MessageBoxButtons.OKCancel) != System.Windows.Forms.DialogResult.OK)

{

e.Cancel = true;

}

}

}

## Form Hàng

Graphical user interface

Description automatically generated

Hình 17 Form Hàng hóa

Form hàng có chức năng cung cấp thông tin về các mặt hàng bao gồm:

- Lưu: Nhập thông tin hàng mới bấm vào lưu để thêm mới vào form

- Sửa: Cập nhật thông tin hàng bị thiếu hoặc sai

- Xóa: Xóa thông tin không cần thiết

- Tìm kiếm: Nhập tên loại hàng cần tìm kiếm

**Các sự kiện trong form**

**Load\_form**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch(Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình kết nối DB");

}

string sQuery = " select \* from HANGHOA ";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds,"HANGHOA");

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

**Lưu**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình kết nối DB");

}

string sMaH = txtMaH.Text;

string sTenH = txtTenH.Text;

string sDonVi = txtDonVi.Text;

string sDonGiaNhap = txtDongia.Text;

// string sDonGiaBan = txtDongiaban.Text;

string sQuery = "insert into HANGHOA values(@MaHH, @TenHH, @DonVi, @DonGia)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHH", sMaH);

cmd.Parameters.AddWithValue("@TenHH", sTenH);

cmd.Parameters.AddWithValue("@DonVi", sDonVi);

cmd.Parameters.AddWithValue("@DonGia", sDonGiaNhap);

// cmd.Parameters.AddWithValue("@DonGiaBan", sDonGiaBan);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công!");

}

catch(Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình thêm mới!");

}

string sQuery1 = "Select \* from HANGHOA";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "HANGHOA");

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

**Cell\_click**

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

txtMaH.Text = dataGridView1.Rows[e.RowIndex].Cells["MaHH"].Value.ToString();

txtTenH.Text = dataGridView1.Rows[e.RowIndex].Cells["TenHH"].Value.ToString();

txtDonVi.Text = dataGridView1.Rows[e.RowIndex].Cells["DonVi"].Value.ToString();

txtDongia.Text = dataGridView1.Rows[e.RowIndex].Cells["DonGia"].Value.ToString();

txtMaH.Enabled = false;

}

**Sửa**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình kết nối DB");

}

string sMaH = txtMaH.Text;

string sTenH = txtTenH.Text;

string sDonVi = txtDonVi.Text;

string sDonGiaNhap = txtDongia.Text;

// string sDonGiaBan = txtDongiaban.Text;

string sQuery = "update HANGHOA set TenH = @TenHH, DonVi =@DonVi, DonGiaNhap=@DonGia where MaH=@MaHH";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHH", sMaH);

cmd.Parameters.AddWithValue("@TenHH", sTenH);

cmd.Parameters.AddWithValue("@DonVi", sDonVi);

cmd.Parameters.AddWithValue("@DonGia", sDonGiaNhap);

// cmd.Parameters.AddWithValue("@DonGiaBan", sDonGiaBan);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công!");

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình cập nhật!");

}

string sQuery1 = "Select \* from HANGHOA";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "HANGHOA");

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

**Xóa**

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

{

DialogResult ret = MessageBox.Show("Bạn có chắc chắn xóa không", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình kết nối DB");

}

string sMaH = txtMaH.Text;

string sQuery = "delete HANGHOA where MaH= @MaHH";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHH", sMaH);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!");

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình xóa!");

}

string sQuery1 = "Select \* from HANGHOA";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "HANGHOA");

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

**Tìm kiếm**

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

{

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Đã xảy ra lỗi trong qúa trình kết nối db");

}

String sQuery = "Select \* from HANGHOA where TenHH like N'%" + txtTimkiem.Text + "%'";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

try

{

adapter.Fill(ds, "HANGHOA");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

## Form Nhà cung cấp

Graphical user interface

Description automatically generated

Hình 19 Form nhà cung cấp

From nhà cung cấp có chức năng quản lý các nhà cung cấp và hàng hóa:

* Lưu: Nhập thông tin nhà cung cấp mới rồi click vào thêm hóa đơn
* Sửa: Cập nhật thông tin bị sai hoặc thiếu
* Xóa: Xóa thông tin hàng
* Tìm kiếm: Nhập tên nhà cung cấp để tìm kiếm
* **Các sự kiện trong form**

**Load\_form**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

String sQuery = "select \* from NHACUNGCAP";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NHACUNGCAP");

dataGridView1.DataSource = ds.Tables["NHACUNGCAP"];

con.Close();

}

**Thêm**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong qúa trình kết nối ");

}

String sMaCC = txtMaCC.Text;

String sTenCC = txtTenCC.Text;

String sDiachi = txtDiachi.Text;

string sSodienthoai = txtSodienthoai.Text;

String sQuery = " insert into NHACUNGCAP values(@MaNCC, @TenNCC, @DiaChi, @Sdt)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaNCC", sMaCC);

cmd.Parameters.AddWithValue("@TenNCC", sTenCC);

cmd.Parameters.AddWithValue("@DiaChi", sDiachi);

cmd.Parameters.AddWithValue("@Sdt", sSodienthoai);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công!", "Thông báo");

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình thêm mới!", "Thông báo");

}

string sQuery1 = "Select \* from NHACUNGCAP ";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NHACUNGCAP ");

dataGridView1.DataSource = ds.Tables["NHACUNGCAP "];

con.Close();

}

**Cell\_click**

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

txtMaCC.Text = dataGridView1.Rows[e.RowIndex].Cells["MaNCC"].Value.ToString();

txtTenCC.Text = dataGridView1.Rows[e.RowIndex].Cells["TenNCC"].Value.ToString();

txtDiachi.Text = dataGridView1.Rows[e.RowIndex].Cells["DiaChi"].Value.ToString();

txtSodienthoai.Text = dataGridView1.Rows[e.RowIndex].Cells["Sdt"].Value.ToString();

txtMaCC.Enabled = false;

}

**Sửa**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong qúa trình kết nối ");

}

String sMaCC = txtMaCC.Text;

String sTenCC = txtTenCC.Text;

String sDiachi = txtDiachi.Text;

String sSodienthoai = txtSodienthoai.Text;

String sQuery = "update NHACUNGCAP set TenCC = @TenNCC, Diachi = @DiaChi, Sodienthoai= @Sdt where MaCC = @MaNCC ";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaNCC", sMaCC);

cmd.Parameters.AddWithValue("@TenNCC", sTenCC);

cmd.Parameters.AddWithValue("@DiaChi", sDiachi);

cmd.Parameters.AddWithValue("@Sdt", sSodienthoai);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật thành công!", "Thông báo");

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình cập nhật!", "Thông báo");

}

string sQuery1 = "Select \* from NHACUNGCAP";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "NHACUNGCAP");

dataGridView1.DataSource = ds.Tables["NHACUNGCAP"];

con.Close();

}

**Xóa**

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

{

DialogResult ret = MessageBox.Show("Bạn có chắc chắn xóa không", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong qúa trình kết nối ");

}

String sMaCC = txtMaCC.Text;

String sQuery = " delete NHACUNGCAP where MaCC = @MaNCC";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("MaCC", sMaCC);

try

{

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!", "Thông báo");

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình xóa!", "Thông báo");

}

string sQuery1 = "Select \* from HANGHOA";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery1, con);

DataSet ds = new DataSet();

adapter.Fill(ds, "HANGHOA");

dataGridView1.DataSource = ds.Tables["HANGHOA"];

con.Close();

}

}

**Tìm kiếm**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong trong qúa trình kết nối");

}

String sQuery = "Select \* from NHACUNGCAP where TenNCC like N'%" + txtTimkiem.Text + "%'";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

try

{

adapter.Fill(ds, "NHACUNGCAP");

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

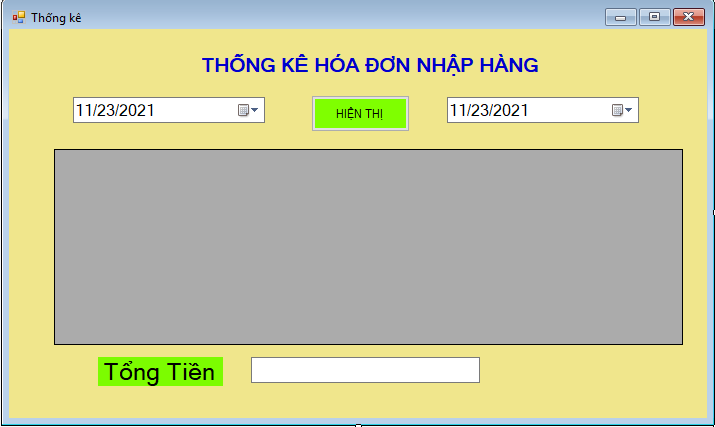
}

dataGridView1.DataSource = ds.Tables["NHACUNGCAP"];

con.Close();

}

## Form thống kê



Hình 20 Form thống kê

From thống kê có chức năng thống kê các đơn nhập hàng:

Giúp người dùng thống kê, tìm kiếm dễ dàng được số lượng hàng hóa và các loại hàng hóa nhập vào theo ngày, tháng, năm.

* **Các sự kiện trong form**

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

{

SqlConnection con = new SqlConnection(sCon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

string sQuery = "select TenHH,TenNCC, SUM(SoLuong)SoLuongNhap, SUM(ThanhTien)TongTien from HANGHOA join NHAP\_CHITIET on HANGHOA.MaHH = NHAP\_CHITIET.MaHH join NHAP on NHAP.MaHDN = NHAP\_CHITIET.MaHDN join NHACUNGCAP on NHACUNGCAP.MaNCC = NHAP.MaNCC where NgayLapHD between @from and @to group by TenHH, TenNCC ";

string sFrom = date1.Value.ToString("yyyy-MM-dd");

string sTo = date2.Value.ToString("yyyy-MM-dd");

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@from", sFrom);

cmd.Parameters.AddWithValue("@to", sTo);

DataTable TK = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(cmd);

adapter.Fill(TK);

dataGridView1.DataSource = TK;

if (TK.Rows.Count > 0)

{

txtTong.Text = (Convert.ToInt32(TK.Compute("SUM (TongTien)", string.Empty)).ToString());

}

else

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

con.Close();

}

}

## Form Hóa đơn

Graphical user interface, timeline

Description automatically generated

Hình 18 Form hóa đơn

Form hóa đơn có chức năng quản lý các đơn hàng nhập:

* Thêm hóa đơn: thêm các hóa đơn khi nhập hàng
* Sửa hóa đơn: sửa các hóa đơn
* Xóa hóa đơn: xóa hóa đơn
* Tìm kiếm: hiển thị các thông tin hóa đơn cần tìm kiếm

**Các sự kiện trong form**

**Load\_form**

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

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

string sQuery1 = "select MaHH, TenHH from HANGHOA";

SqlDataAdapter adapter1 = new SqlDataAdapter(sQuery1, con);

DataSet ds1 = new DataSet();

adapter1.Fill(ds1, "HANGHOA");

cbTenH.DataSource = ds1.Tables["HANGHOA"];

cbTenH.ValueMember = "MaHH";

cbTenH.DisplayMember = "TenHH";

string sQuery2 = "select TenNCC, MaNCC from NHACUNGCAP";

SqlDataAdapter adapter2 = new SqlDataAdapter(sQuery2, con);

DataSet ds2 = new DataSet();

adapter2.Fill(ds2, "NHACUNGCAP");

cbMaCC.DataSource = ds2.Tables["NHACUNGCAP"];

cbMaCC.DisplayMember = "TenNCC";

cbMaCC.ValueMember = "MaNCC";

con.Close();

isLoad = 1;

}

**SelectedIndexChanged**

private void cbTenH\_SelectedIndexChanged(object sender, EventArgs e)

{

if(isLoad != 0)

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

string sQuery1 = "select DonGia from HANGHOA where MaHH = " +

cbTenH.SelectedValue.ToString();

SqlDataAdapter adapter1 = new SqlDataAdapter(sQuery1, con);

DataSet ds1 = new DataSet();

adapter1.Fill(ds1, "HANGHOA");

txtDongia.Text = ds1.Tables["HANGHOA"].Rows[0][0].ToString();

con.Close();

}

**Thêm**

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

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

int iMaHD = Convert.ToInt16(txtMaHD.Text);

string sNgayNhap = daNgaynhap.Value.ToString("yyyy-MM-dd");

// string sGioNhap = daGionhap.Value.ToString("h:mm:ss");

string sMaCungCap = cbMaCC.Text;

string sTongtien = txtTongtien.Text;

int iSoLuong = Convert.ToInt16(txtSoluong.Text);

string sQuery = " insert into NHAP(MaHDN, NgayLapHD, MaNCC, TongTien) values(@MaHDN, @NgayLapHD, @ManCC, @TongTien)";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHDN", iMaHD);

cmd.Parameters.AddWithValue("@NgayLapHD", sNgayNhap);

// cmd.Parameters.AddWithValue("@GioNhap", sGioNhap);

cmd.Parameters.AddWithValue("@TongTien", sTongtien);

cmd.Parameters.AddWithValue("@MaNCC", sMaCungCap);

try

{

cmd.ExecuteNonQuery();

// MessageBox.Show("Thêm mới thành công!");

}

catch(Exception ex)

{

MessageBox.Show(ex.ToString());

}

int iCount = DataGridView1.Rows.Count;

for (int i = 0; i < iCount - 1; i++)

{

string sQuery1 = "insert into NHAP\_CHITIET values(@MaHDN, @MaHH, @SoLuong, @ThanhTien)";

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd1.Parameters.AddWithValue("@MaHDN", iMaHD);

cmd1.Parameters.AddWithValue("@MaHH", DataGridView1.Rows[i].Cells[0].Value);

cmd1.Parameters.AddWithValue("@SoLuong", DataGridView1.Rows[i].Cells[3].Value);

cmd1.Parameters.AddWithValue("@ThanhTien", DataGridView1.Rows[i].Cells[4].Value);

{

MessageBox.Show("Thêm mới thành công");

}

con.Close();

}

}

**Sửa**

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

{

if (DataGridView1.SelectedRows.Count == 1) // ktra mot lan sua chi duoc sua 1 mat hang

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

}

}

**Xóa**

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

{

{

string sMaHDN = "";

if (txtMaHD.Text == "")

{

}

else

sMaHDN = txtMaHD.Text;

DialogResult ret = MessageBox.Show("Bạn có chắc xóa không!", "Thông báo", MessageBoxButtons.OKCancel);

if (ret == DialogResult.OK)

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

string sQuery = " delete NHAP\_CHITIET where MaHDN = @MaHDN";

string sQuery1 = " delete NHAP where MaHDN = @MaHDN";

SqlCommand cmd = new SqlCommand(sQuery, con);

cmd.Parameters.AddWithValue("@MaHDN", sMaHDN);

SqlCommand cmd1 = new SqlCommand(sQuery1, con);

cmd1.Parameters.AddWithValue("@MaHDN", sMaHDN);

try

{

cmd.ExecuteNonQuery();

cmd1.ExecuteNonQuery();

MessageBox.Show("Xóa thành công!", "Thông báo");

txtDongia.Text = "";

txtMaHD.Text = "";

txtTongtien.Text = "";

txtThanhtien.Text = "";

txtSoluong.Text = "";

txtTimkiem.Text = "";

}

catch (Exception)

{

MessageBox.Show("Đã xảy ra lỗi trong quá trình xóa!", "Thông báo!");

}

con.Close();

}

}

**Tìm kiếm**

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

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

string sQuery = " select MaHDN, NgayLapHD, MaNCC, TongTien from NHAP where MaHDN = ' " + txtTimkiem.Text + "'";

SqlCommand cmd = new SqlCommand(sQuery, con);

SqlDataReader ds = cmd.ExecuteReader();

if (ds.Read() == true)

{

txtMaHD.Text = ds["MaHDN"].ToString();

txtTongtien.Text = ds["TongTien"].ToString();

daNgaynhap.Value = Convert.ToDateTime(ds["NgayLapHD"].ToString());

}

else

MessageBox.Show("Đơn hàng không tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

ds.Close();

////2. Lay du lieu tu HDNHAP hiển thị lên datagr

string SQuery1 = "select NHAP\_CHITIET.MaHH, TenHH, HANGHOA.DonGia, NHAP\_CHITIET.SoLuong, ThanhTien from NHAP\_CHITIET inner join HANGHOA on NHAP\_CHITIET.MaHH = HANGHOA.MaHH where NHAP\_CHITIET.MaHDN = @MaHDN";

SqlCommand cmd1 = new SqlCommand(SQuery1, con);

cmd1.Parameters.AddWithValue("@MaHDN", txtTimkiem.Text);

SqlDataAdapter adapter = new SqlDataAdapter(cmd1);

DataSet ts = new DataSet();

adapter.Fill(ts, "HDCHITIET");

int iRows = ts.Tables["HDCHITIET"].Rows.Count;

for (int i = 0; i < iRows; i++)

{

DataGridView1.Rows.Add(ts.Tables["HDCHITIET"].Rows[i][0], ts.Tables["HDCHITIET"].Rows[i][1], ts.Tables["HDCHITIET"].Rows[i][2], ts.Tables["HDCHITIET"].Rows[i][3], ts.Tables["HDCHITIET"].Rows[i][4]);

}

con.Close();

}

**Cell\_click**

private void DataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

cbTenH.Text = DataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

txtSoluong.Text = DataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception ex)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

//lay hang khi biet mah va truyen vao gd

string sQuery = "select \* from HANGHOA where TenHH = '" + cbTenH.Text + "'";

SqlDataAdapter adapter = new SqlDataAdapter(sQuery, con);

DataSet ds = new DataSet();

adapter.Fill(ds,"HANGHOA");

DataTable dt = ds.Tables["HANGHOA"];

DataRow r = dt.Rows[0]; // lay hang dau tien trong data tble

// hien thi tenh va dongia

cbTenH.Text = r["TenHH"].ToString();

txtDongia.Text = r["DonGia"].ToString();

con.Close();

}

**Add\_click**

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

{

SqlConnection con = new SqlConnection(scon);

try

{

con.Open();

}

catch (Exception)

{

MessageBox.Show("Xảy ra lỗi trong quá trình kết nối DB");

}

if (txtSoluong.Text != "")

{

int iSoLuong = Convert.ToInt32(txtSoluong.Text);

int iDonGia = Convert.ToInt32(txtDongia.Text);

int iThanhtien = iSoLuong \* iDonGia;

txtThanhtien.Text = iThanhtien.ToString();

int iCout = DataGridView1.Rows.Count;

int ktra = 0;

if (iCout > 1)

{

for (int i = 0; i < iCout - 1; i++)

{

if (DataGridView1.Rows[i].Cells[0].Value.ToString() == cbTenH.SelectedValue.ToString())

{

DataGridView1.Rows[i].Cells[3].Value = iSoLuong + Convert.ToInt32(DataGridView1.Rows[i].Cells[3].Value);

DataGridView1.Rows[i].Cells[4].Value = Convert.ToInt32(DataGridView1.Rows[i].Cells[3].Value) \* iDonGia;

ktra = 1;

}

}

}

if (ktra == 0)

{

DataGridView1.Rows.Add(cbTenH.SelectedValue, cbTenH.Text, iDonGia, iSoLuong, iThanhtien);

}

long iTongtien;

if (txtTongtien.Text == "")

iTongtien = 0;

else

iTongtien = Convert.ToInt32(txtTongtien.Text);

iTongtien = iTongtien + iThanhtien;

txtTongtien.Text = iTongtien.ToString();

}

}

**Clear\_click**

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

{

txtMaHD.Text = "";

cbMaCC.Text = "";

txtSoluong.Text = "";

txtThanhtien.Text = "";

cbTenH.Text = "";

txtDongia.Text = "";

int indexOfRows = DataGridView1.CurrentCell.RowIndex;

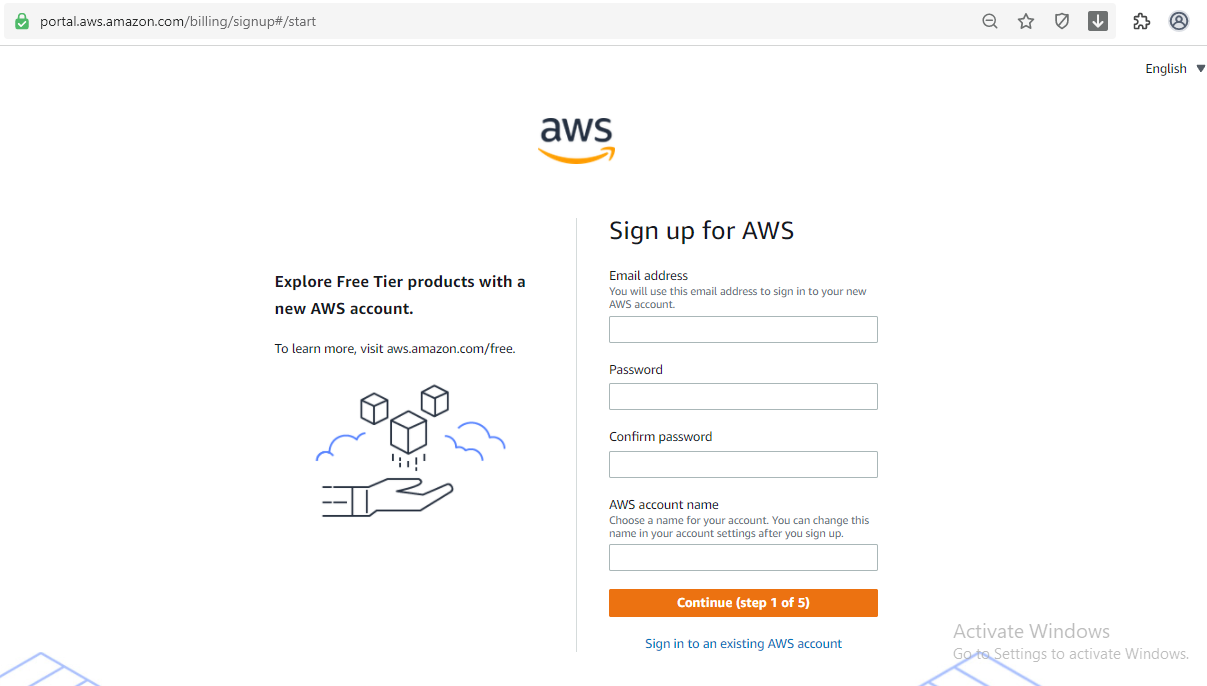
DataGridView1.Rows.RemoveAt(indexOfRows);

}

# VIII. Phương án giải quyết vấn đề dữ liệu lớn:

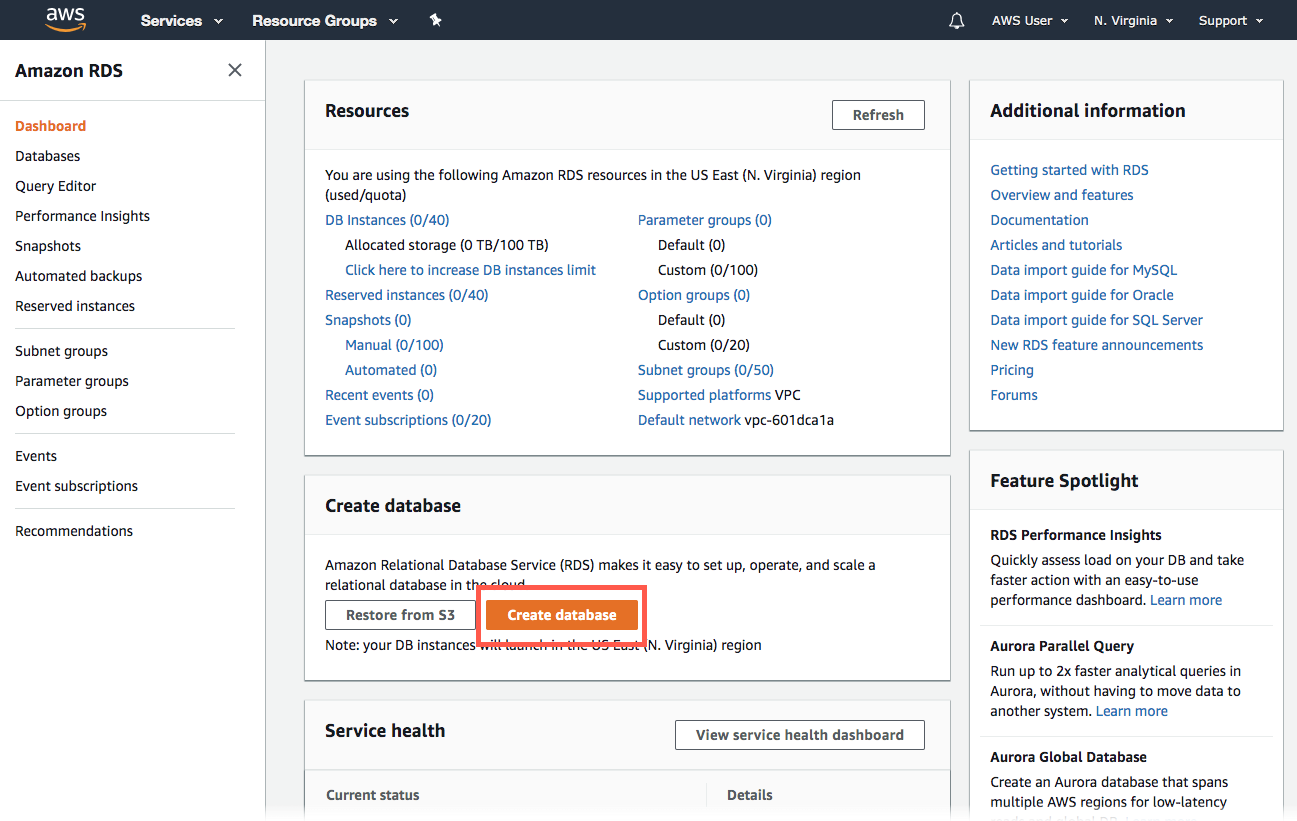
Khái niệm điện toán đám mây (*cloud computing*) chắc hẳn không còn xa lạ đối với các doanh nghiệp ngày nay. Khi các doanh nghiệp gặp vấn đề về dữ liệu quá lớn, không đủ lưu trữ trong 1 ổ đĩa, thì điện toán đám mây mây là giải pháp phù hợp để lưu trữ database. Và một trong những nền tảng điện toán đám mây tiêu biểu đó là Amazon Relational Database Service (Amazon) .RDS) .Amazon Relational Database Service (Amazon). Amazon RDS) giúp dễ dàng định cấu hình, sử dụng và mở rộng cơ sở dữ liệu quan hệ trên đám mây

Bước 1: Tạo tài khoản



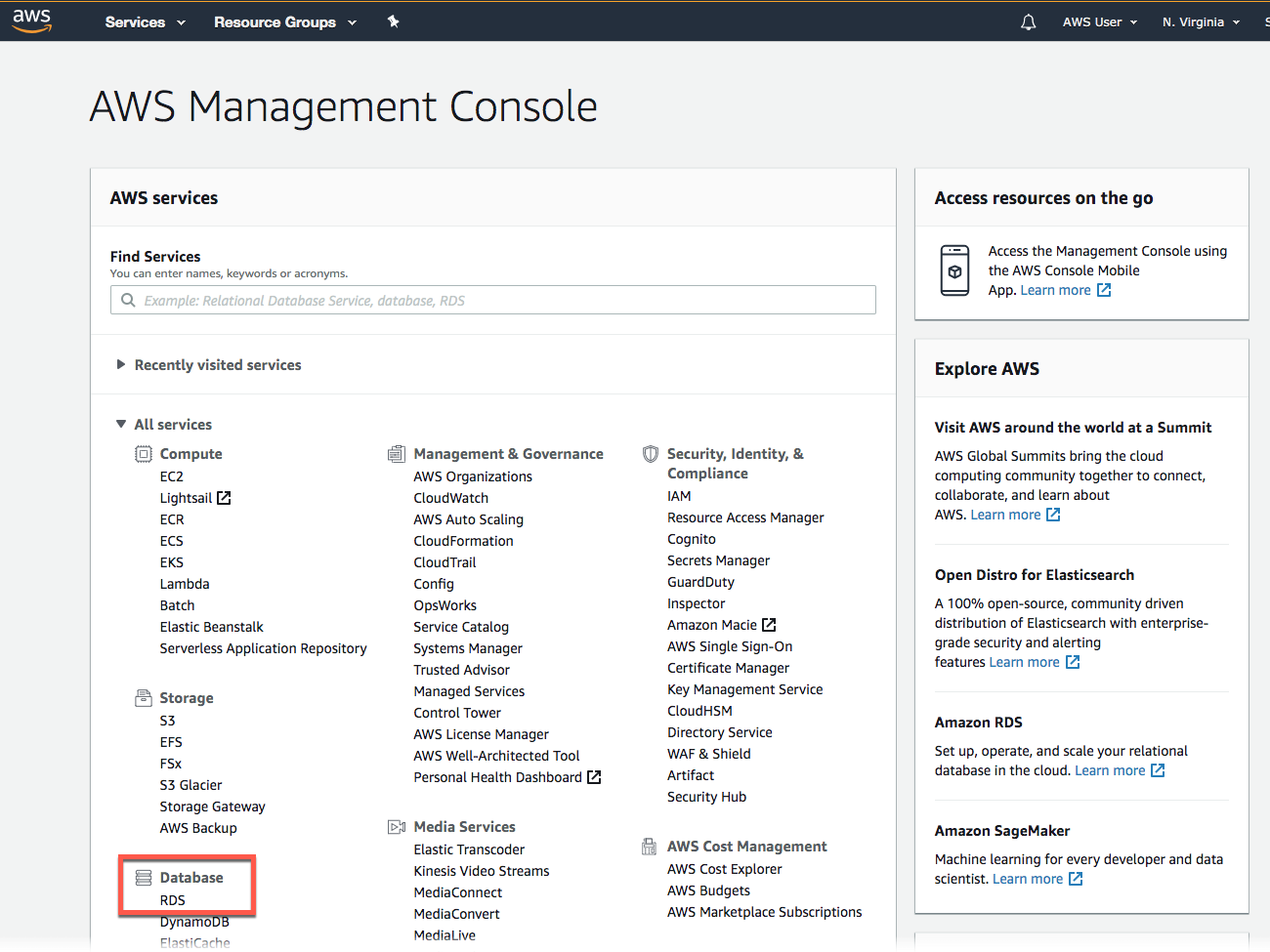
Hình 20 Tạo tài khoản AWS

Bước 2: Tạo cơ sở dữ liệu



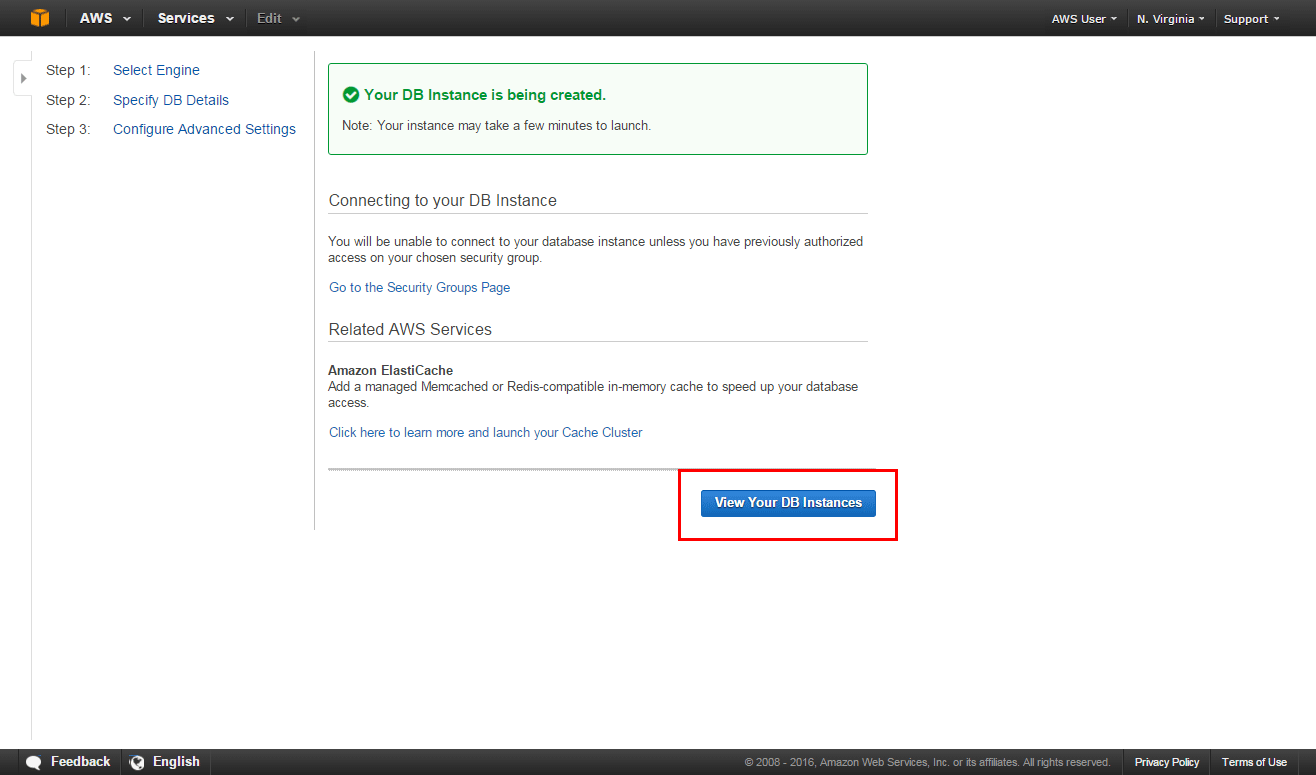
Hình 21 Tạo cơ sở dữ liệu

Bước 3: Vào bảng điều khiển



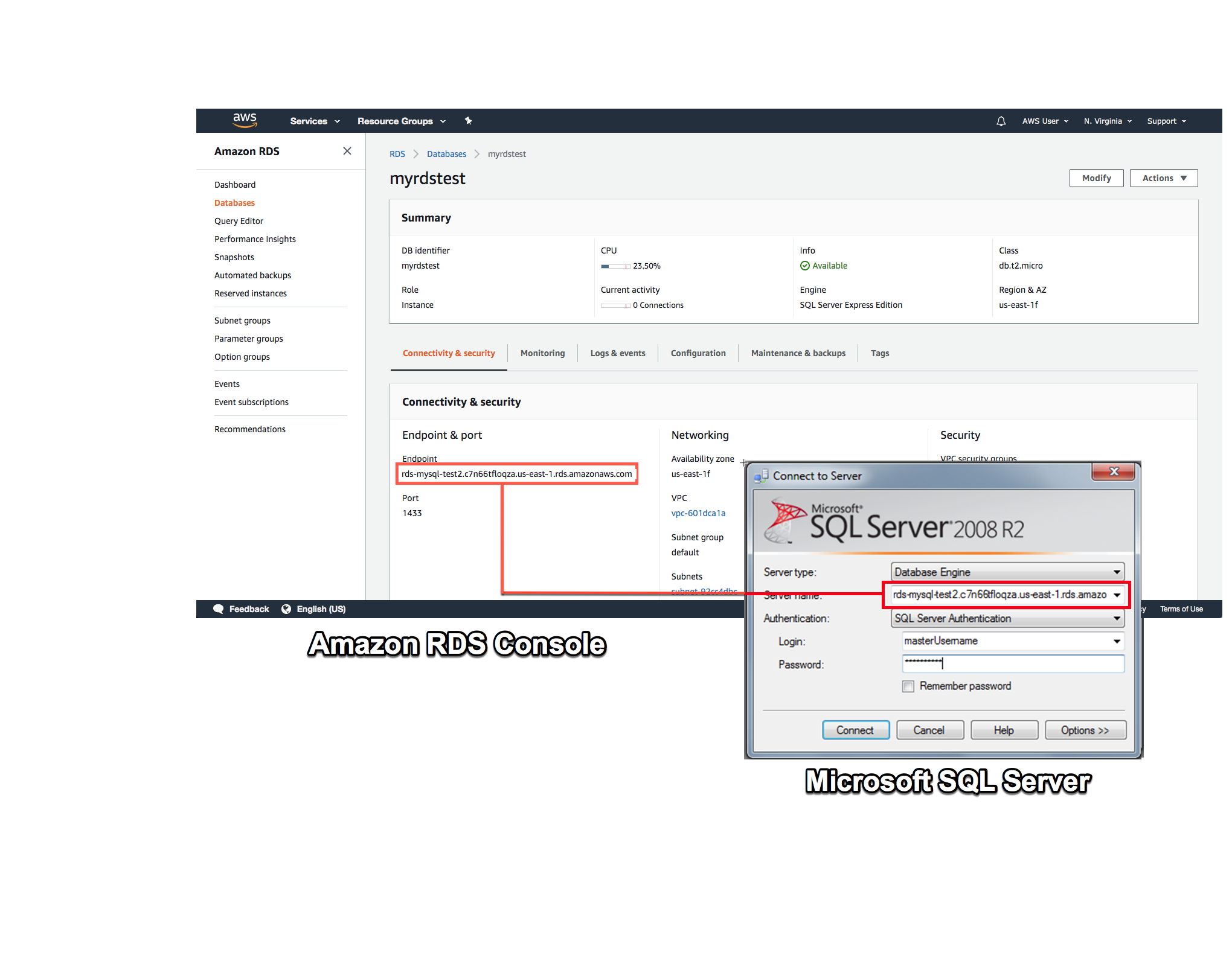
Hình 22 Vào bảng điều khiển

Bước 4: Tạo thành công



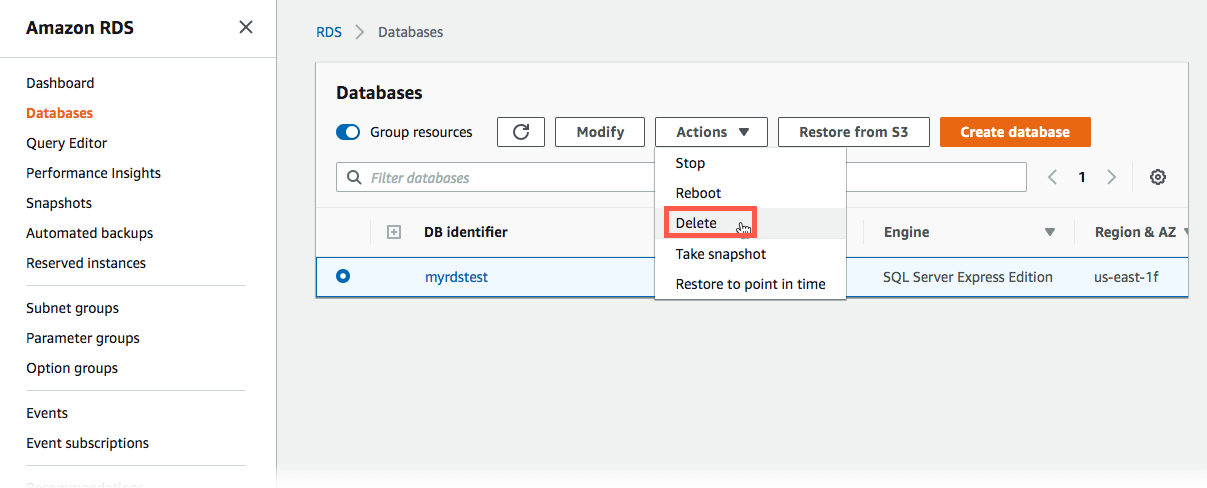
Hình 23 Tạo thành công

Bước 5: Kết nối với cơ sở dữ liệu Microsoft SQL Server



Hình 24 Kết nối Microsoft SQL Server

Bước 6: Xóa phiên cơ sở dữ liệu



Hình 25 Xóa phiên cơ sở dữ liệu

# IX.   Đề phòng phương án bị tấn công bằng SQL Injection:

Để DB NHAPHANG có thể bảo vệ bản thân trước sự tấn công bằng SQL Injection, hãy kết hợp sử dụng các phương pháp sau đây:

* Phương pháp 1: ở phương này nhóm sẽ sử dụng store procedure để đề phòng sự tấn công thay vì sử dụng câu lệnh SELECT
* Phương pháp 2: Phân quyền cho account Database. Hãy tạo và gán quyền truy cập cho account ấy. Hạn chế sử dụng account root. Điều này giúp phòng ngừa được việc hacker xâm nhập vào được SQL để đọc dữ liệu từ các bảng hoặc thay đổi dữ liệu
* Phương pháp 3: Thường xuyên sao lưu backup: Hãy backup thường xuyên để dữ liệu luôn có thể được khôi phục nếu có sự đe dọa xâm nhập, tàn phá dữ liệu bởi các hacker

# TÀI LIỆU THAM KHẢO

1. <https://aws.amazon.com/vi/getting-started/hands-on/create-microsoft-sql-db/>
2. <http://3i.com.vn/index.php/tai-lieu-erp/tip-huong-dan/sql-server-agent-va-cach-tao-job-active-monitor/>
3. <https://soncrm.wordpress.com/2017/02/19/tao-job-tu-dong-backup-database-va-chuyen-file-qua-ftp/>