

```

create PROC [dbo].[SP_TAOLOGIN]
    @LGNAME VARCHAR(50),
    @PASS VARCHAR(50),
    @USERNAME VARCHAR(50),
    @ROLE VARCHAR(50)
AS
    DECLARE @RET INT
    EXEC @RET= SP_ADDLOGIN @LGNAME, @PASS, 'QLDSV'
    IF (@RET =1) -- LOGIN NAME BI TRUNG
        RETURN 1

    EXEC @RET= SP_GRANTDBACCESS @LGNAME, @USERNAME
    IF (@RET =1) -- USER NAME BI TRUNG
    BEGIN
        EXEC SP_DROPLOGIN @LGNAME
        RETURN 2
    END
    EXEC sp_addrolemember @ROLE, @USERNAME
    IF @ROLE= 'ADMIN_TRUONG' OR @ROLE= 'ADMIN_KHOA'
    BEGIN
        EXEC sp_addsrvrolemember @LGNAME, 'SecurityAdmin'
        EXEC sp_addsrvrolemember @LGNAME, 'ProcessAdmin'
    END
    RETURN 0 -- THANH CONG

```

```

CREATE PROC [dbo].[Xoa_Login]
    @LGNAME VARCHAR(50),
    @USRNAME VARCHAR(50)

```

```

AS
    EXEC SP_DROPUSER @USRNAME
    EXEC SP_DROPLOGIN @LGNAME

```

```

CREATE VIEW V_NHANVIEN
SELECT  MANV, HO, TEN, HO + '' + TEN AS HOTEN
FROM    dbo.NHANVIEN
ORDER BY TEN, HO

```

* Program :

```

namespace QLDSV_PHANTAN
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        public static SqlConnection conn = new
SqlConnection();
        public static String connstr;
        public static SqlDataAdapter da;
    }
}

```

```

        public static SqlDataReader myReader;
        public static String servername = "";
        public static String servername1 = "THU-PC\\"
TINTIN";
        public static String servername2 = "THU-PC\\SUSU";
        public static String username;
        public static String password;
        public static String database = "QLDSV";
        public static String mlogin;
        public static String mGroup;
        public static String mHoten;

        public static int KetNoi()
        {
            if (Program.conn != null && Program.conn.State
== ConnectionState.Open) Program.conn.Close();
            try
            {
                Program.connstr = "Data Source=" +
Program.servername + ";Initial Catalog=" + Program.database
+ ";User ID=" +
                Program.mlogin + ";password=" +
Program.password;
                Program.conn.ConnectionString =
Program.connstr;
                Program.conn.Open();
                return 1;
            }

            catch (Exception e)
            {
                MessageBox.Show("Lỗi kết nối cơ sở dữ
liệu.\nBạn xem lại user name và password.\n " + e.Message,
"", MessageBoxButtons.OK);
                return 0;
            }
        }

        public static SqlDataReader
ExecSqlDataReader(String cmd, String connectionstring)
        {
            SqlDataReader myreader;
            //Program.conn = new
SqlConnection(connectionstring);

            SqlCommand sqlcmd = new SqlCommand();
            sqlcmd.Connection = Program.conn;
            sqlcmd.CommandText = cmd;
            sqlcmd.CommandType = CommandType.Text;

```

```

        if (Program.conn.State ==
ConnectionState.Closed) Program.conn.Open();
        try
        {
            myreader = sqlcmd.ExecuteReader(); return
myreader;
        }
        catch (SqlException ex)
        {
            Program.conn.Close();
            MessageBox.Show(ex.Message);
            return null;
        }
    }
}

```

```

    public static DataTable ExecSqlQuery(String cmd,
String connectionString)
    {
        DataTable dt1 = new DataTable();
        conn = new SqlConnection(connectionString);
        da = new SqlDataAdapter(cmd, conn);
        da.Fill(dt1);
        return dt1;
    }
}

```

```

    public static int ExecSqlNonQuery(String cmd,
String connectionString)
    {
        conn = new SqlConnection(connectionString);
        SqlCommand Sqlcmd = new SqlCommand();
        Sqlcmd.Connection = conn;
        Sqlcmd.CommandText = cmd;
        Sqlcmd.CommandType = CommandType.Text;
        Sqlcmd.CommandTimeout = 300;
        if (conn.State == ConnectionState.Closed)
conn.Open();
        try
        {
            Sqlcmd.ExecuteNonQuery(); conn.Close();

return 1;
        }
        catch (SqlException ex)
        {
            MessageBox.Show(ex.Message);
            conn.Close();
            return 0;
        }
    }
}

```

```

    }
    public static void SetEnableOfButton(Form frm,
Boolean Active)
    {
        foreach (Control ctl in frm.Controls)
            if ((ctl) is Button)
                ctl.Enabled = Active;
    }

    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new frmMenu());
    }
}

```

```
CREATE PROCEDURE Tim_Hoten  
@manv int  
AS  
select ho, ten from nhanvien where manv = @manv  
GO
```