

SQL QUERY

```
create database db_student
```

```
use db_student
```

```
create table info(  
id int identity(1,1) primary key clustered,  
stud_no as 'Stud_no' + Right('0000'+ CAST(id as varchar(5)),5) persisted,  
fname varchar (50),  
lname varchar (50),  
bday Date,  
s_address varchar (50),  
phone varchar (50),  
email varchar (50)  
)
```

```
create procedure sp_save  
@fname varchar(50),  
@lname varchar(50),  
@bday Date,  
@s_address varchar(50),  
@phone varchar (50),  
@email varchar(50)  
as  
insert into info values (@fname,@lname,@bday,@s_address,@phone, @email)
```

```
create procedure sp_update  
@id int,  
@fname varchar(50),  
@lname varchar(50),  
@bday Date,  
@s_address varchar(50),  
@phone varchar (50),  
@email varchar(50)  
as  
update info  
set fname = @fname, lname = @lname, bday = @bday, s_address = @s_address, phone = @phone,  
email = @email  
where id=@id
```

```
create procedure sp_delete  
@id int  
as  
delete info  
where id=@id
```

```
create procedure sp_view  
as  
select  
id as 'ID',  
stud_no as 'STUD_NO',  
fname as 'FIRST NAME',  
lname as 'LAST NAME',  
bday as 'BIRTH DATE',
```

```
DATEDIFF(YY,bday,GETDATE()) as 'AGE',
s_address as 'ADDRESS',
phone as 'PHONE',
email as 'EMAIL'
from info
```

```
create procedure sp_search
@key varchar(max)
as
select
id as 'ID',
stud_no as 'STUD_NO',
fname as 'FIRST NAME',
lname as 'LAST NAME',
bday as 'BIRTH DATE',
DATEDIFF(YY,bday,GETDATE()) as 'AGE',
s_address as 'ADDRESS',
phone as 'PHONE',
email as 'EMAIL'
from info
where
stud_no like '%' +@key+'%'
or fname like '%' +@key+'%'
or lname like '%' +@key+'%'
or bday like '%' +@key+'%'
or s_address like '%' +@key+'%'
or phone like '%' +@key+'%'
or email like '%' +@key+'%'
```

VSC CODES

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Mahinay_Student_Information_System
{
    public partial class Form1 : Form
    {
        int id;
        DataClasses1DataContext db = new DataClasses1DataContext();
        public Form1()
        {
            InitializeComponent();
            db = new DataClasses1DataContext();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            RefreshDataGridView();
        }
        private void Clear()
        {
            textFname.Text = "";
            textLname.Text = "";
            dateTimePicker1.Text = "";
            textAddress.Text = "";
            textPhone.Text = "";
            textEmail.Text = "";
        }

        private void RefreshDataGridView()
        {
            dataGridView1.DataSource = db.sp_view();
        }

        //SAVING
        private void button1_Click(object sender, EventArgs e)
        {
            if (Validator())
            {
                DateTime textBday = DateTime.Parse(dateTimePicker1.Text);
                db.sp_save(textFname.Text, textLname.Text, textBday,
textAddress.Text, textPhone.Text, textEmail.Text);
                MessageBox.Show("Successfully saved", "Save");
                Clear();
                RefreshDataGridView();
            }
        }
    }
}
```

```

    }
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    id = int.Parse(dataGridView1.CurrentRow.Cells[0].Value.ToString());
    textFname.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textLname.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
    dateTimePicker1.Text =
dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textAddress.Text = dataGridView1.CurrentRow.Cells[6].Value.ToString();
    textPhone.Text = dataGridView1.CurrentRow.Cells[7].Value.ToString();
    textEmail.Text = dataGridView1.CurrentRow.Cells[8].Value.ToString();
}

//Updating
private void buttonUpdate_Click(object sender, EventArgs e)
{
    if (id > 0 && Validator())
    {
        db.sp_update(id, textFname.Text, textLname.Text,
dateTimePicker1.Value, textAddress.Text, textPhone.Text, textEmail.Text);
        MessageBox.Show("Successfully Updated", "Update");
        Clear();
        RefreshDataGridView();
    }
    else
    {
        MessageBox.Show("Please select a record and ensure all fields are
filled correctly.", "Update");
    }
}

//DELETING
private void button3_Click(object sender, EventArgs e)
{
    if (id > 0)
    {
        db.sp_delete(id);
        MessageBox.Show("Successfully Deleted", "Delete");
        Clear();
        RefreshDataGridView();
    }
    else
    {
        MessageBox.Show("Please select a record to delete.", "Delete");
    }
}

//SEARCHING
private void textBox3_TextChanged(object sender, EventArgs e)
{
    dataGridView1.DataSource = db.sp_search(textBox3.Text);
}

//VALIDATOR
private bool Validator()

```

```

{
    if (string.IsNullOrEmpty(textFname.Text) ||
        string.IsNullOrEmpty(textLname.Text) ||
        string.IsNullOrEmpty(textAddress.Text) ||
        string.IsNullOrEmpty(textPhone.Text) ||
        string.IsNullOrEmpty(textEmail.Text))
    {
        MessageBox.Show("All fields are required.", "Validation Error");
        return false;
    }

    string namePattern = @"^[a-zA-Z\s]+$";
    if (!Regex.IsMatch(textFname.Text, namePattern) ||
        !Regex.IsMatch(textLname.Text, namePattern))
    {
        MessageBox.Show("First Name and Last Name should contain only
letters.", "Validation Error");
        return false;
    }

    string emailPattern = @"^\\w+([-+.'\\w+)*@\\w+([-.]\\w+)*\\.\\w+([-.]\\w+)*$";
    if (!Regex.IsMatch(textEmail.Text, emailPattern))
    {
        MessageBox.Show("Invalid email format.", "Validation Error");
        return false;
    }

    string phonePattern = @"^\\d{11}$";
    if (!Regex.IsMatch(textPhone.Text, phonePattern))
    {
        MessageBox.Show("Invalid phone number format. Please enter 11
digits.", "Validation Error");
        return false;
    }

    return true;
}
}

```