Crud Operation using SQLite with C sharp windows form

We are going to discuss about how to Crud operation using SQLite with c sharp windows form.

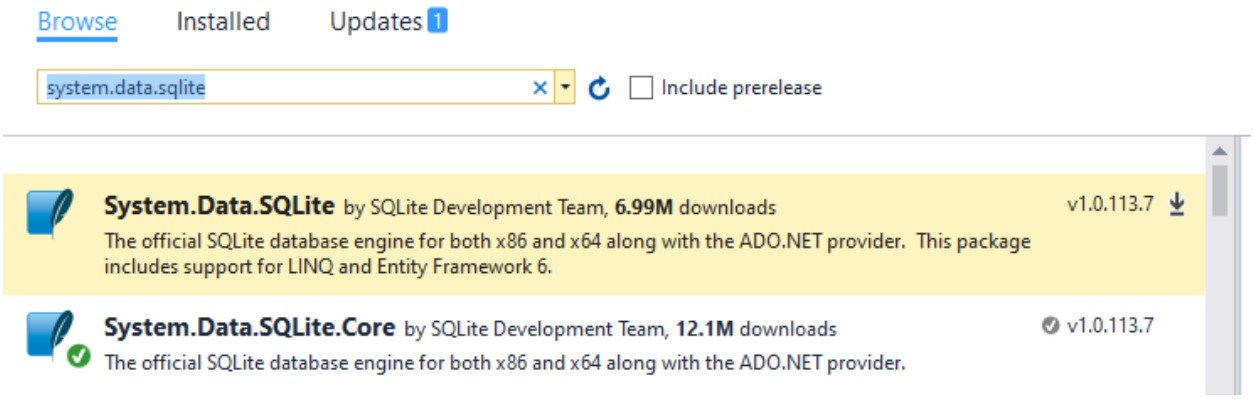
Firstly, we have to keep our SQLite database file within our project's bin folder, Database named "SupplierMgt.db" and create table which is "tblSupplier" using SQLite .

```
CREATE TABLE "tblSupplier" (
        "ID"    INTEGER,
        "Code"          TEXT,
        "Name"          TEXT,
        "Mobile"        TEXT,
        "Email"         TEXT,
        "Address"       TEXT,
        "CreatedDate"           TEXT,
        PRIMARY KEY("ID" AUTOINCREMENT)
)
```

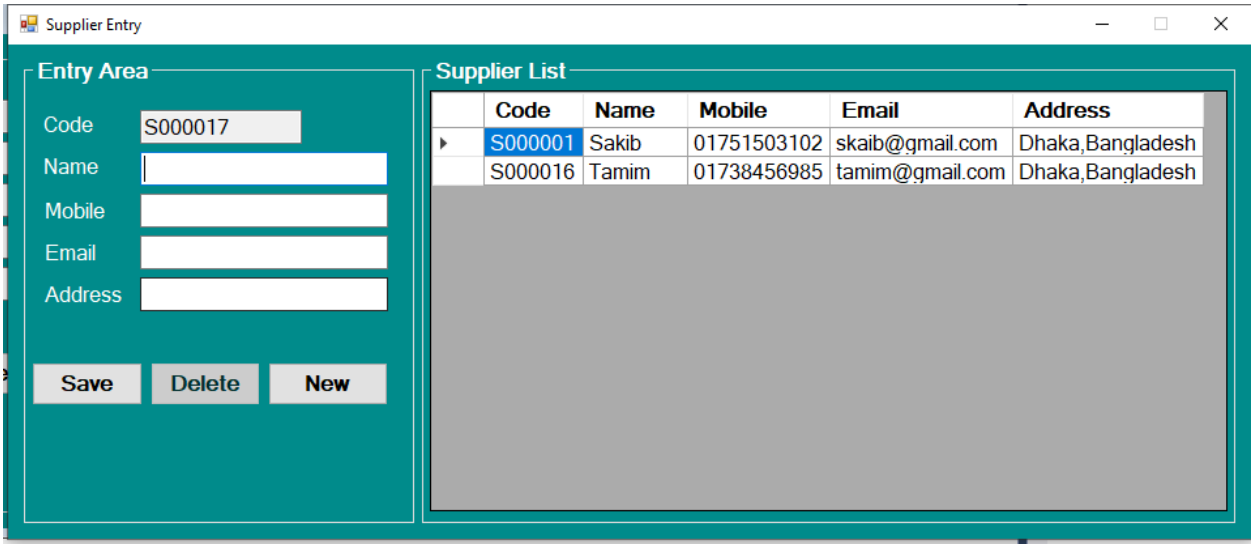
Step 1: web config file will be like that below

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="Default" connectionString="Data Source=.\SupplierMgt.db; Version=3;"
providerName="System.Data.SqlClient" />
  </connectionStrings>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.1" />
    </startup>
</configuration>
```

Step 2: We have to add reference in our project which is System.Data.SQlite.Core dll file from NuGet Manager



Step 3: We have designed a supplier entry form named "SupplierFrm" which UI given below



Step 4: We have to add a class which is used for making connection with SQLite database named `DataAccess`

```
public  class DataAccess
    {

        static string _ConnectionString =
ConfigurationManager.ConnectionStrings["Default"].ConnectionString;



        static SQLiteConnection _Connection = null;
        public static SQLiteConnection Connection
```

```
    {
        get
        {
            if (_Connection == null)
            {
                _Connection = new SQLiteConnection(_ConnectionString);
                _Connection.Open();

                return _Connection;
            }
            else if (_Connection.State != System.Data.ConnectionState.Open)
            {
                _Connection.Open();

                return _Connection;
            }
            else
            {
                return _Connection;
            }
        }
    }

    public static DataSet GetDataSet(string sql)
    {
        SQLiteCommand cmd = new SQLiteCommand(sql, Connection);
        SQLiteDataAdapter adp = new SQLiteDataAdapter(cmd);

        DataSet ds = new DataSet();
        adp.Fill(ds);
        Connection.Close();

        return ds;
    }

    public static DataTable GetDataTable(string sql)
    {
        Console.WriteLine(sql);
        DataSet ds = GetDataSet(sql);

        if (ds.Tables.Count > 0)
            return ds.Tables[0];
        return null;
    }

    public static int ExecuteSQL(string sql)
    {
        SQLiteCommand cmd = new SQLiteCommand(sql, Connection);
        return cmd.ExecuteNonQuery();
    }
}
```

Step 5: Finally we have to keep below codes within “SupplierFrm.cs” class. But make sure that all events must be tag with UI controls

```
public partial class SupplierFrm : Form
    {
        public SupplierFrm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

            LoadSupplierList();
            Clear();


        }

        private string SupplierCodeGenerate()
        {
            string pvCode = "S";
            int IncrementSerial = 0;
            try
            {
                string sqlQuery = "SELECT Max(ID) as ID FROM tblSupplier";
                DataTable DtTable = DataAccess.GetDataTable(sqlQuery);
                string pvSerial = DtTable.Rows[0]["ID"].ToString();

                // it has been used for ID incremnet
                if (pvSerial == "")
                {
                    pvSerial = "1";
                }
                else
                {
                   IncrementSerial = int.Parse(pvSerial) + 1;
                    pvSerial = IncrementSerial.ToString();
                }


                // It has been Used for Code generation's formating

                if (pvSerial.Length == 1)
                {
                    pvCode = pvCode + "00000" + pvSerial;
                }

                if (pvSerial.Length == 2)
                {
                    pvCode = pvCode + "0000" + pvSerial;
                }
                if (pvSerial.Length == 3)
                {
                    pvCode = pvCode + "000" + pvSerial;
                }
                if (pvSerial.Length == 4)
```

```
                {
                    pvCode = pvCode + "00" + pvSerial;
                }
                if (pvSerial.Length == 5)
                {
                    pvCode = pvCode + "0" + pvSerial;
                }

            }
            catch (Exception ex)
            {

                MessageBox.Show(ex.Message.ToString(), "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
            return pvCode;


        }

        private void LoadSupplierList()
        {
            string sqlQuery = "SELECT Code, Name, Mobile, Email, Address FROM
tblSupplier";
            DataTable DtTable = DataAccess.GetDataTable(sqlQuery);
            if (DtTable.Rows.Count >0)
            {
                grdSupplierList.DataSource = DtTable;
                grdSupplierList.AutoResizeColumns();
                grdSupplierList.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
            }
        }

        private void btnSave_Click(object sender, EventArgs e)
        {

            try
            {
                if (txtName.Text.Trim() == "")
                {
                    MessageBox.Show("Please Input Name", "Alert", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                    txtName.Focus();
                    return;
                }
                if (txtMobile.Text.Trim() == "")
                {
                    MessageBox.Show("Please Input Mobile", "Alert", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                    txtMobile.Focus();
                    return;
                }
                else
                {
                    if (btnSave.Text == "Save") // For Save Operation
                    {
                        string CurrentDate = DateTime.Now.ToString("yyyy-MM-dd");
```

```
                        string sqlQuery = "INSERT INTO tblSupplier (Code, Name, Mobile,
Email, Address,CreatedDate) VALUES('" + txtSupplierCode.Text.Trim() + "','" +
txtName.Text.Trim() + "','" + txtMobile.Text.Trim() + "','" + txtEmail.Text.Trim() +
"','" + txtAddress.Text.Trim() + "','" + CurrentDate + "')";
                        int result = DataAccess.ExecuteSQL(sqlQuery);
                        if (result > 0)
                        {
                            MessageBox.Show("Supplier Saved Successfully.", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                            LoadSupplierList();
                            Clear();

                        }
                    }
                    else // Update Operation
                    {

                        string sqlQuery = "UPdate tblSupplier SET
Name='"+txtName.Text.Trim()+
"',Mobile='"+txtMobile.Text.Trim()+"',Email='"+txtEmail.Text.Trim()+"',Address='"+txtAddr
ess.Text.Trim()+"' WHERE Code='" + txtSupplierCode.Text.Trim() + "'";
                        int result = DataAccess.ExecuteSQL(sqlQuery);
                        if (result > 0)
                        {
                            MessageBox.Show("Supplier Updated Successfully.", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                            LoadSupplierList();
                            Clear();

                        }
                    }
                }
            }
            catch (Exception ex)
            {

                MessageBox.Show(ex.Message.ToString(), "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void Clear()
        {
            txtSupplierCode.Text = SupplierCodeGenerate();
            txtName.Text = "";
            txtMobile.Text = "";
            txtEmail.Text = "";
            txtAddress.Text = "";
            btnSave.Text = "Save";
            btnDelete.Enabled = false;

        }

        private void btnDelete_Click(object sender, EventArgs e)
        {
            try
            {
```

```
                DialogResult = MessageBox.Show("Do you want to delete?", "Delete",
MessageBoxButtons.YesNo, MessageBoxIcon.Information);
                if (DialogResult == DialogResult.Yes)
                {
                    string sqlQuery = "DELETE FROM tblSupplier WHERE
Code='"+txtSupplierCode.Text.Trim()+"'";
                    int result = DataAccess.ExecuteSQL(sqlQuery);
                    if (result > 0)
                    {
                        MessageBox.Show("Supplier Deleted Successfully.", "Delete",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        LoadSupplierList();
                        Clear();

                    }

                }
                else
                {
                    txtName.Focus();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message.ToString(), "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void btnNew_Click(object sender, EventArgs e)
        {
            try
            {
                Clear();
                txtName.Focus();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message.ToString(), "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void grdSupplierList_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
        {
            try
            {
                btnSave.Text = "Update";
                btnDelete.Enabled = true;

                int rowIndex = e.RowIndex;
                if (rowIndex >= 0)
                {
                    DataGridViewRow row = grdSupplierList.Rows[rowIndex];
                    string Code = row.Cells[0].Value.ToString();
                    if (Code != "")
                    {
```

```
                        string sqlQuery = "SELECT * FROM tblSupplier  WHERE Code='" +
Code + "'";
                        DataTable DtTable = DataAccess.GetDataTable(sqlQuery);
                        txtSupplierCode.Text = DtTable.Rows[0]["Code"].ToString();
                        txtName.Text= DtTable.Rows[0]["Name"].ToString();
                        txtMobile.Text = DtTable.Rows[0]["Mobile"].ToString();
                        txtEmail.Text = DtTable.Rows[0]["Email"].ToString();
                        txtAddress.Text = DtTable.Rows[0]["Address"].ToString();
                        txtName.Focus();
                    }
                }
                else
                {
                    btnSave.Text = "Save";
                    btnDelete.Enabled = false;
                }
            }
            catch (Exception ex)
            {

                MessageBox.Show(ex.Message.ToString(), "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void txtName_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                txtMobile.Focus();
            }
        }

        private void txtMobile_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                txtEmail.Focus();
            }
        }

        private void txtEmail_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                txtAddress.Focus();
            }
        }

        private void txtAddress_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                btnSave.Focus();
            }
        }
    }
```