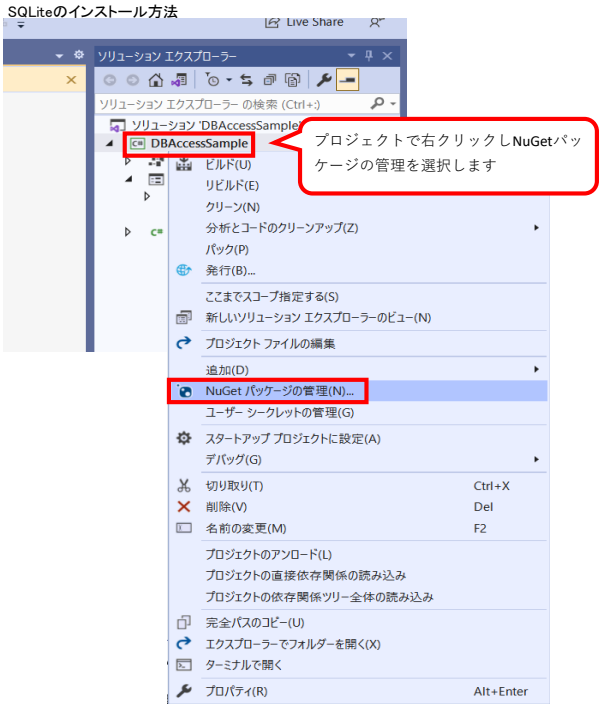
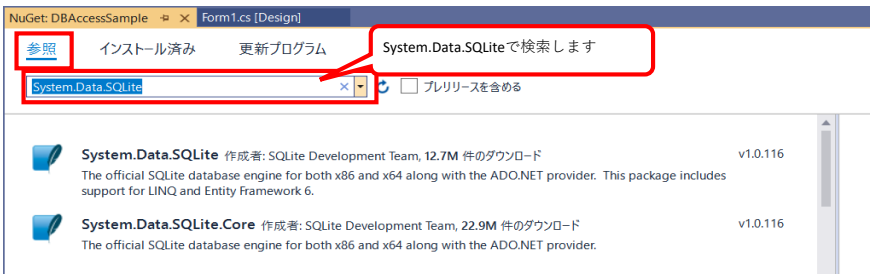
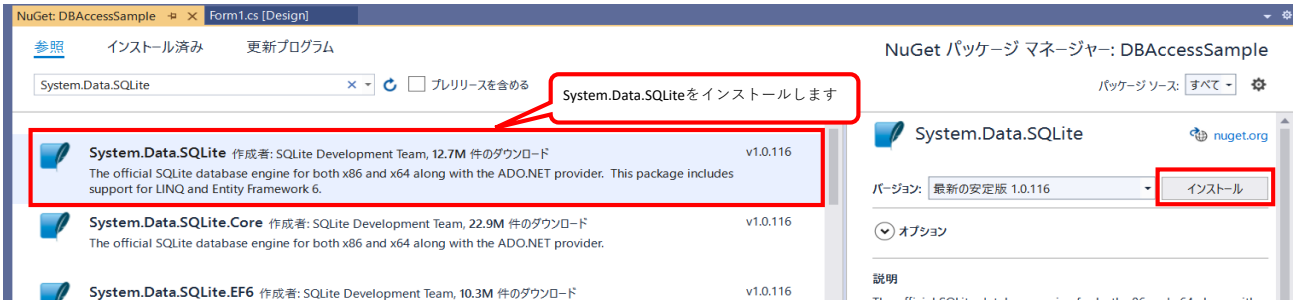
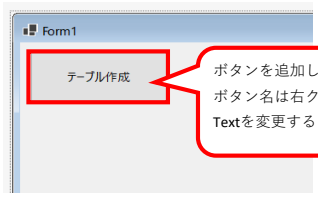
SQLiteのインストール方法







Formに部品を設置してDBにアクセスします



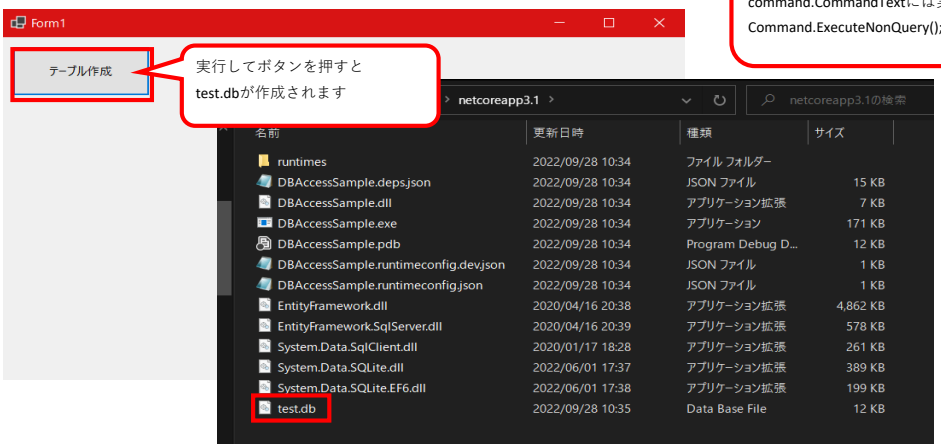
ボタンを追加してダブルクリックでコードを表示します
ボタン名は右クリック→プロパティからプロパティを表示して
Textを変更すると変わります

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SQLite;

namespace DBAccessSample {
    3 個の参照
    public partial class Form1 : Form {
        1 個の参照
        public Form1() {
            InitializeComponent();
        }

        1 個の参照
        private void button1_Click(object sender, EventArgs e) {
            // コネクションを開いてテーブル作成して閉じる
            using (var con = new SQLiteConnection("Data Source=test.db")) {
                con.Open();
                using (SQLiteCommand command = con.CreateCommand()) {
                    command.CommandText =
                        "create table t_product(CD INTEGER  PRIMARY KEY AUTOINCREMENT, productname TEXT, price INTEGER)";
                    command.ExecuteNonQuery();
                }
                con.Close();
            }
        }
    }
}
```

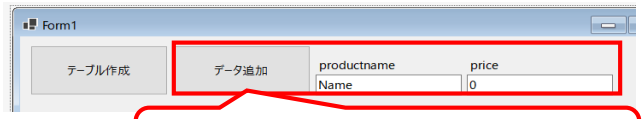
Data Source=で指定した名前がDBのファイル名になります

t_productテーブルを作成します

con.Open();でDBを接続しcon.Close();でDB接続を切断します
使い終わったら必ず接続を切断しましょう
command.CommandTextには実行するSQL文を入れます
Command.ExecuteNonQuery();でSQLを実行します



実行してボタンを押すと
test.dbが作成されます



データ追加用ボタンとラベル、テキストぼっくを追加します
データ追加ボタンをダブルクリックしてコードを追記します

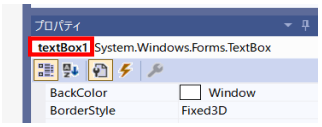
```
1 個の参照
private void button2_Click(object sender, EventArgs e) {
    using (SQLiteConnection con = new SQLiteConnection("Data Source=test.db")) {
        con.Open();
        using (SQLiteTransaction trans = con.BeginTransaction()) {
            SQLiteCommand cmd = con.CreateCommand();
            // インサート
            cmd.CommandText = "INSERT INTO t_product (productname, price) VALUES (@Product, @Price)";
            // パラメータセット
            cmd.Parameters.Add("Product", System.Data.DbType.String);
            cmd.Parameters.Add("Price", System.Data.DbType.Int64);
            // データ追加
            cmd.Parameters["Product"].Value = textBox1.Text;
            cmd.Parameters["Price"].Value = int.Parse(textBox2.Text);
            cmd.ExecuteNonQuery();
            // コミット
            trans.Commit();
        }
    }
}
```

@をつけることで後述のParametersでセットした値をプログラム
実行時に付加してSQLを実行できます

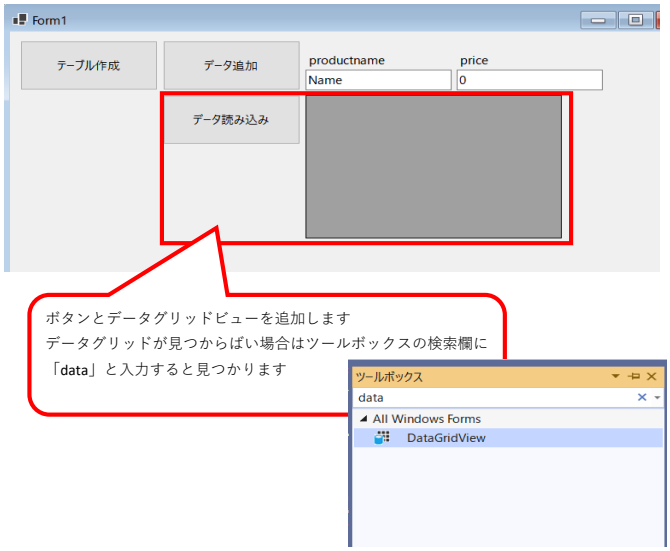
ParametersにProductとPriceを追加します

textBoxに入力されている文字列をParametersに設定します

textBox1とtextBox2は追加したテキストボックスの名称です
デザイナー画面のプロパティから確認できます

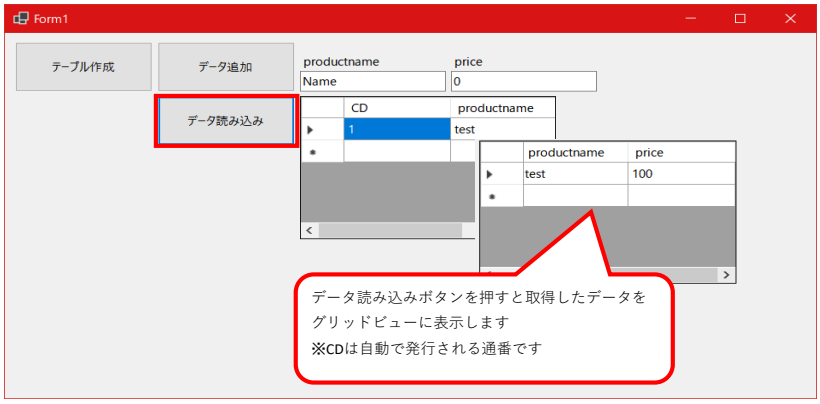


trans.commit();でDBの変更を確定します



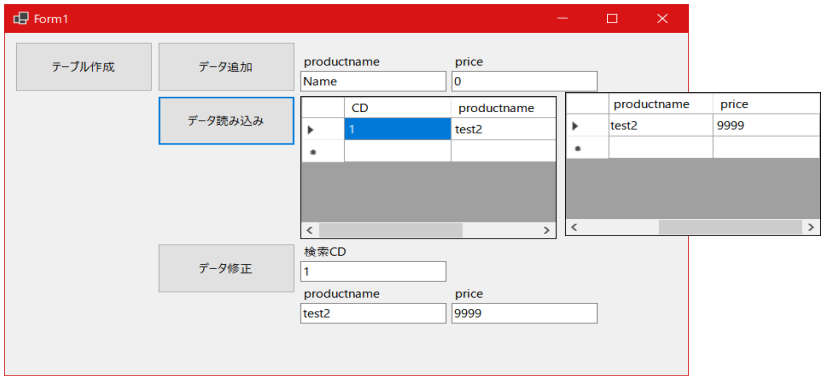

```
1 個の参照
private void button3_Click(object sender, EventArgs e) {
    using (SQLiteConnection con = new SQLiteConnection("Data Source=test.db")) {
        // DataTableを生成します。
        var dataTable = new DataTable();
        // SQLの実行
        var adapter = new SQLiteDataAdapter("SELECT * FROM t_product", con);
        adapter.Fill(dataTable);
        dataGridView1.DataSource = dataTable;
    }
}
```

SQLiteDataAdapterでSelect文を発行し、結果をdataTableに取得します
その後データグリッドビューに設定します
https://learn.microsoft.com/ja-jp/dotnet/api/system.data.sqlclient.sqldataadapter?view=dotnet-plat-ext-6.0



```
private void button4_Click(object sender, EventArgs e) {
    using (SQLiteConnection con = new SQLiteConnection("Data Source=test.db")) {
        con.Open();
        using (SQLiteTransaction trans = con.BeginTransaction()) {
            SQLiteCommand cmd = con.CreateCommand();
            // インサート
            cmd.CommandText = "UPDATE FROM t_product set productname = @Product, price = @Price WHERE CD = @Cd";
            // パラメータセット
            cmd.Parameters.Add("Product", System.Data.DbType.String);
            cmd.Parameters.Add("Price", System.Data.DbType.Int64);
            cmd.Parameters.Add("Cd", System.Data.DbType.Int64);
            // データ追加
            cmd.Parameters["Product"].Value = textBox3.Text;
            cmd.Parameters["Price"].Value = int.Parse(textBox4.Text);
            cmd.Parameters["Cd"].Value = int.Parse(textBox5.Text);
            cmd.ExecuteNonQuery();
            // コミット
            trans.Commit();
        }
    }
}
```

UPDATE文を作ることもできます




```
private void button5_Click(object sender, EventArgs e) {
    using (SQLiteConnection con = new SQLiteConnection("Data Source=test.db")) {
        con.Open();
        using (SQLiteTransaction trans = con.BeginTransaction()) {
            SQLiteCommand cmd = con.CreateCommand();
            // インサート
            cmd.CommandText = "DELETE FROM t_product WHERE CD = @Cd;";
            // パラメータセット
            cmd.Parameters.Add("Cd", System.Data.DbType.Int64);
            // データ削除
            cmd.Parameters["Cd"].Value = int.Parse(textBox6.Text);
            cmd.ExecuteNonQuery();
            // コミット
            trans.Commit();
        }
    }
}
```

DELETE文を作ることもできます


```
private void button6_Click(object sender, EventArgs e) {
    // コネクションを開いてテーブル削除して閉じる
    using (var con = new SQLiteConnection("Data Source=test.db")) {
        con.Open();
        using (SQLiteCommand command = con.CreateCommand()) {
            command.CommandText =
                "drop table t_product";
            command.ExecuteNonQuery();
        }
        con.Close();
    }
}
```

テーブルを削除するときはdrop tableを使います