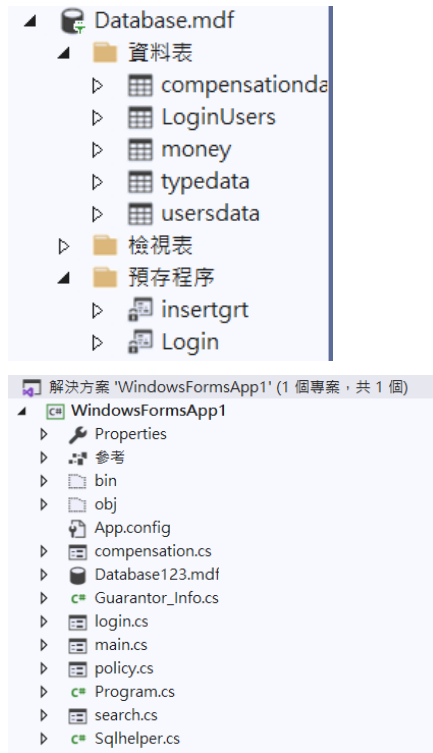


保險公司之資料管理者使用的系統

組員：CBF108046 曾怡禎、CBF108058 林珊羽

使用：資料表*5、預存程序*2、LINQ*1



- **第一部分(login.cs)**

首先，一開始執行的是登入系統的畫面(login.cs)。這一部分是使用預存程序(Login)去檢視帳號(textBox1)跟密碼(textBox2)的值是否存在於**第一張**資料表(LoginUsers)中。



```

//預存程序
SqlCommand cmd = new SqlCommand();
cmd.CommandText = "Login"; //預存程序名稱
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.Add(new SqlParameter("@acco", SqlDbType.NVarChar));
cmd.Parameters["@acco"].Value = textBox1.Text; //帳號
cmd.Parameters.Add(new SqlParameter("@pass", SqlDbType.NVarChar));
cmd.Parameters["@pass"].Value = textBox2.Text; //密碼
SqlDataAdapter da = new SqlDataAdapter();
da.SelectCommand = cmd;
da.SelectCommand.Connection = cn;
DataSet ds = new DataSet();
da.Fill(ds);

```

按關閉這個按鈕，就會關閉整個系統，`Application.Exit()`，其實用 `Close` 也可以，只是如果到其他的 form 再回來這個 form 按關閉系統，只會關掉畫面，可是實際上還是正在執行中。

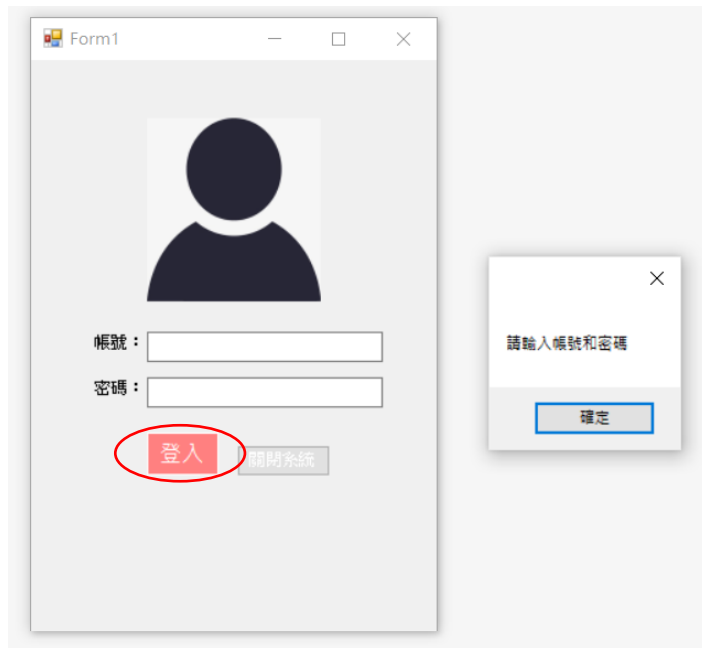


```

private void button2_Click(object sender, EventArgs e) //關閉系統
{
    Application.Exit(); //關閉系統
    //this.Close();
}

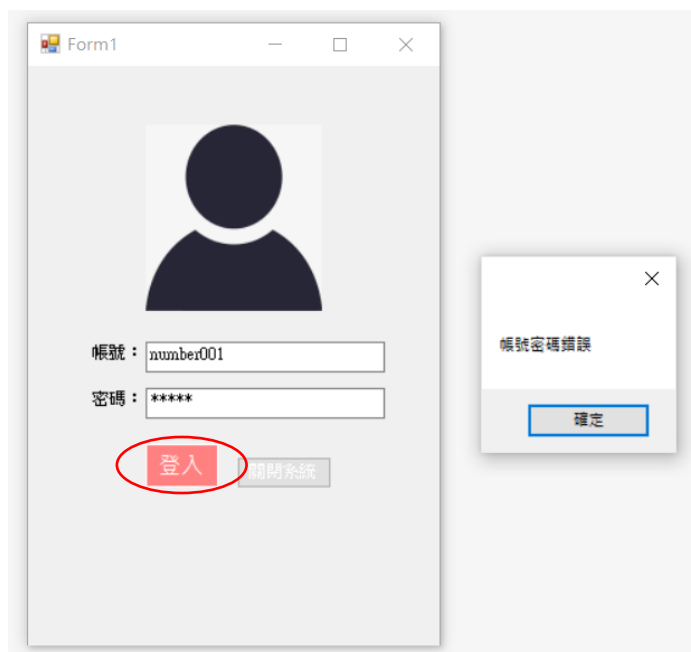
```

如果帳號(textBox1)跟密碼(textBox2)沒輸入值就按登入的話，會顯示”請輸入帳號和密碼”的訊息。



```
if (textBox1.Text == "" && textBox2.Text == "") //判斷輸入欄是否為空值
{
    MessageBox.Show("請輸入帳號和密碼"); //輸出請輸入帳號和密碼
}
```

如果帳號(textBox1)跟密碼(textBox2)輸入的值不存在於資料表(LoginUsers)中，會顯示”帳號密碼錯誤”的訊息。



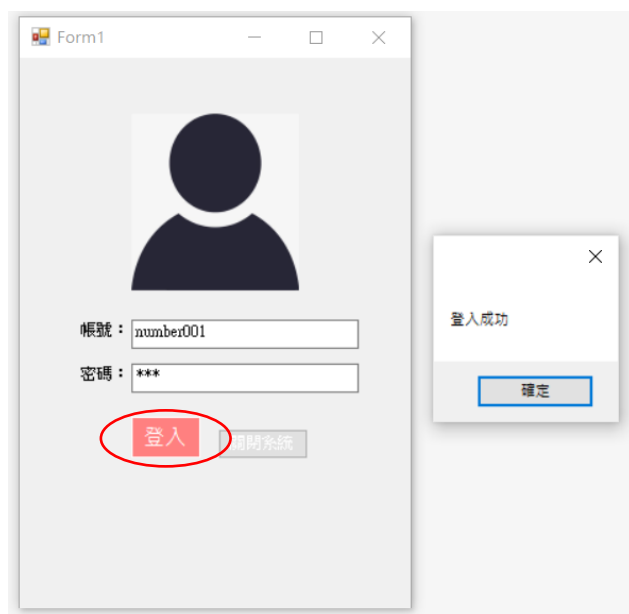
```
MessageBox.Show("帳號密碼錯誤"); //輸出帳號密碼錯誤
```

並將密碼(textBox2)清空，以方便重新輸入。



`textBox2.Text = "";` //將密碼欄清空重新輸入

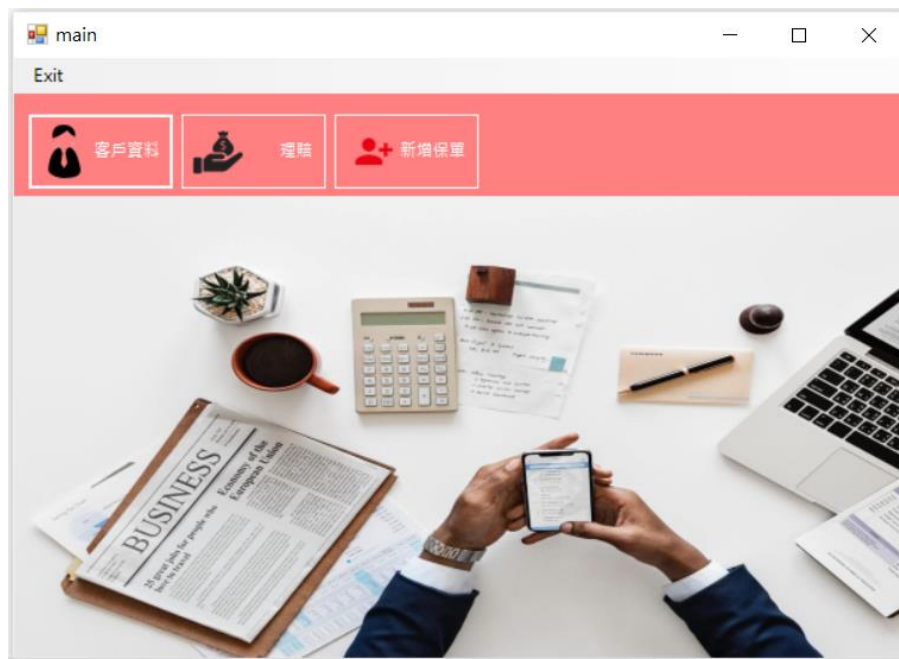
如果帳號(textBox1)跟密碼(textBox2)輸入的值被判斷存在於資料表(LoginUsers)中，會顯示”登入成功”訊息，然後隱藏目前這個 form，顯示下一個 form 的畫面。



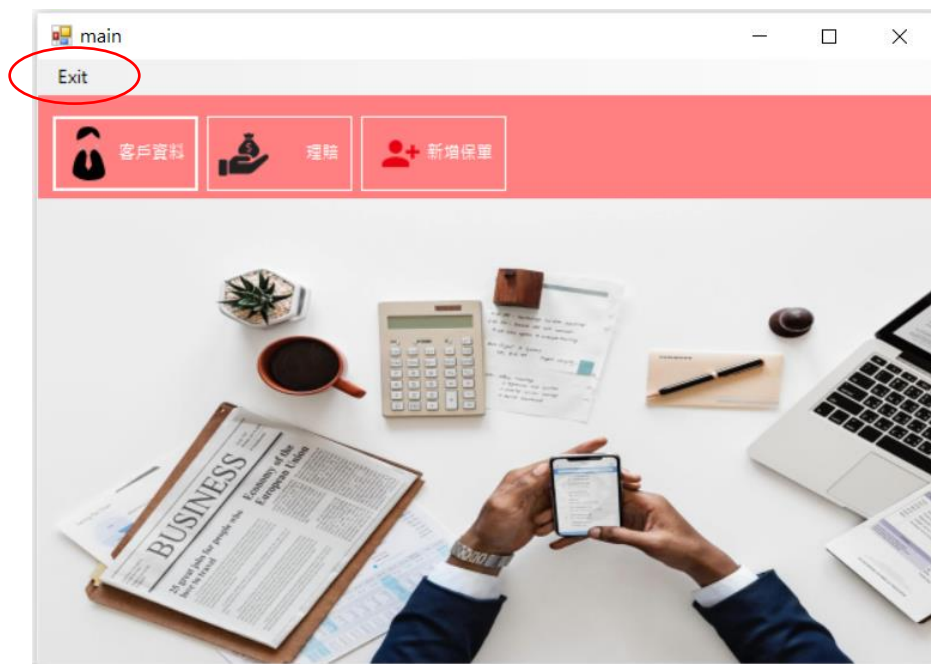
```
int count = ds.Tables[0].Rows.Count; //判斷資料庫中是否有此用戶
if (count == 1) //如果資料庫有這筆記錄
{
    MessageBox.Show("登入成功"); //輸出登入成功
    main two = new main();
    two.Show(); //顯示第二個介面main.cs
    this.Hide(); //隱藏第一個介面login.cs
}
```

- **第二部分(main.cs)**

登入之後，顯示這個畫面(main.cs)。

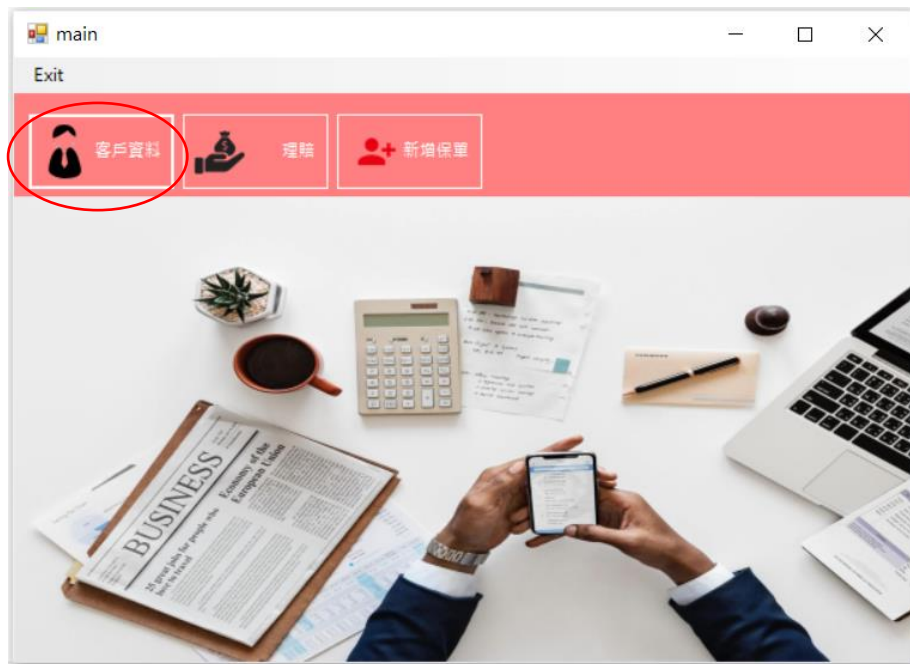


這裡我做了一個登出 Exit，使用 **menuStrip 控制項**，可以將登出釘置在畫面的最上面，按登出則會回到剛剛登入的那個畫面(login.cs)。



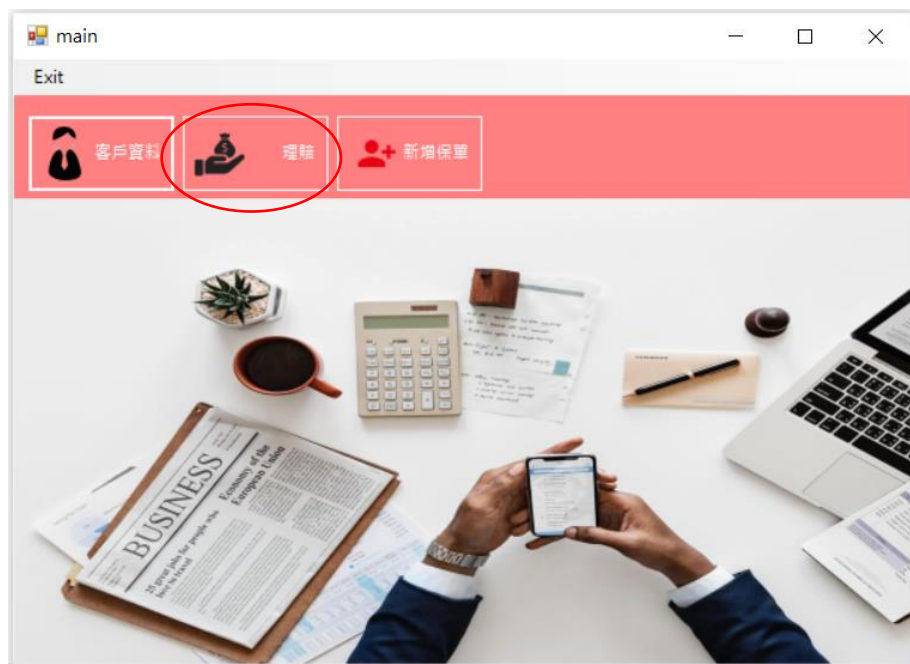
```
private void exitToolStripMenuItem_Click(object sender, EventArgs e) //登入
{
    login one = new login(); //登出返回到第一個介面 顯示login.cs
    one.Show();
    this.Hide(); //隱藏第二個介面main.cs
}
```

點擊客戶資料按鈕會顯示對應的 form(search.cs)。



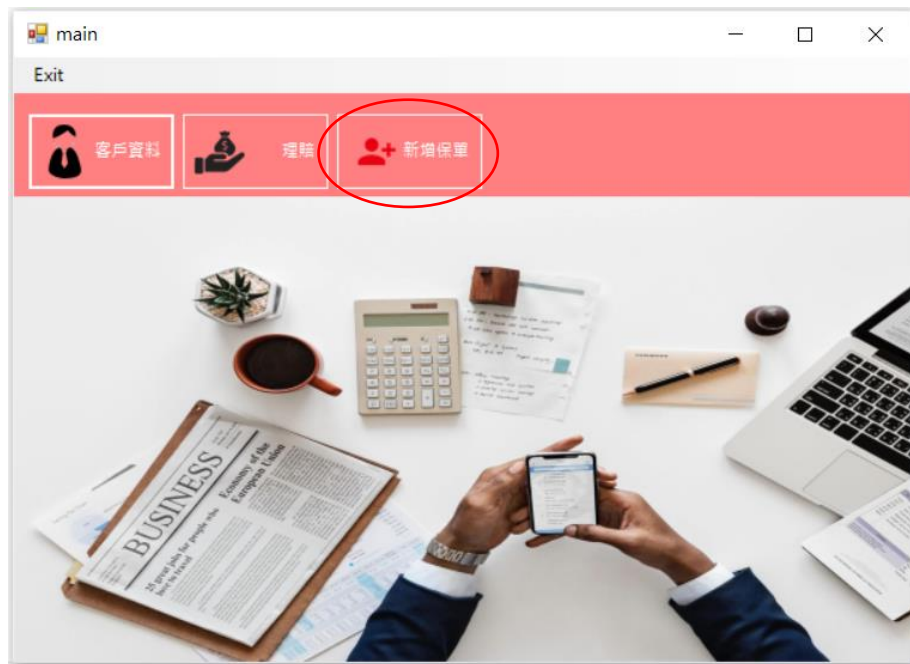
```
private void button1_Click(object sender, EventArgs e) //客戶資料
{
    search three = new search(); //顯示第三個介面search.cs
    three.Show();
}
```

點擊理賠按鈕會顯示對應的 form(compensation.cs)。



```
private void button3_Click(object sender, EventArgs e) //理賠
{
    compensation five = new compensation(); //顯示第五個介面compensation.cs
    five.Show();
}
```

點擊新增保單按鈕會顯示對應的 form(policy.cs)。

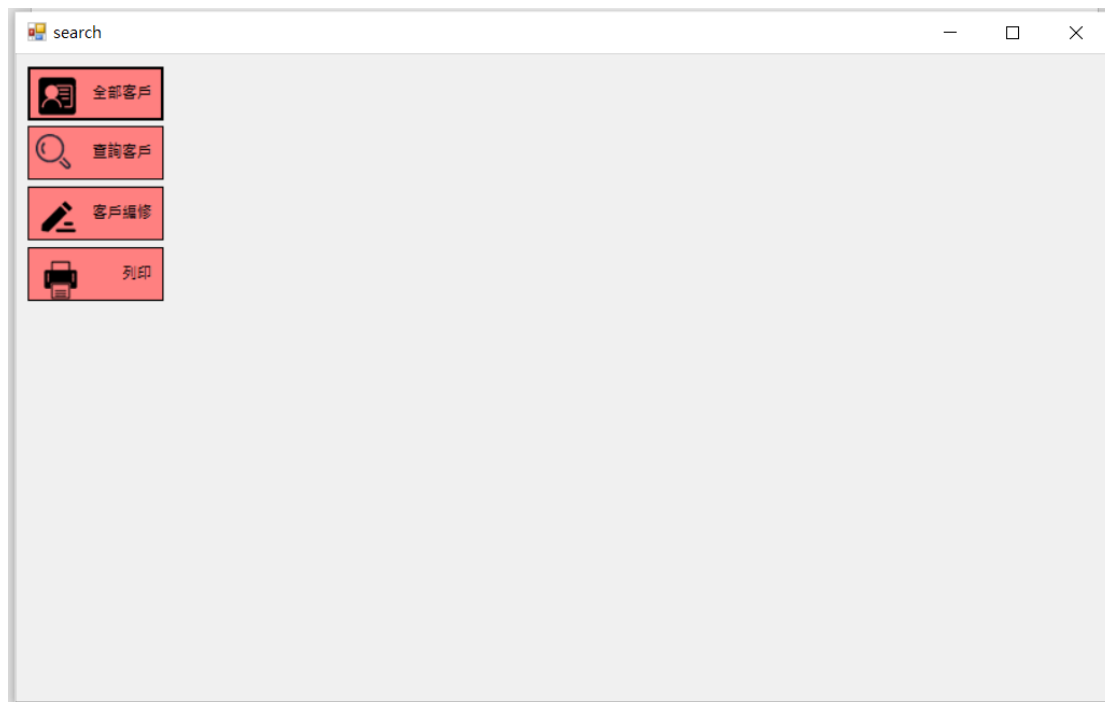


```
private void button2_Click(object sender, EventArgs e) //新增保單
{
    policy four = new policy(); //顯示第四個介面policy.cs
    four.Show();
}
```

這四個部分都是運用 `Show()` 跟 `Hide()`。

- **第三部分(search.cs)**

點擊客戶資料按鈕顯示這個畫面(search.cs)。這裡點擊按鈕顯示的畫面，都是在同一個 form 裡面，所以有用到 `Visible=false` 先隱藏所有 groupBox 控制項。



```
groupBox2.Visible = false; //將全部客戶(客戶編修)的groupBox2資料框隱藏
groupBox1.Visible = false; //將查詢客戶的groupBox1查詢框隱藏
```

點擊全部客戶按鈕顯示資料框，使用 `groupBox` 控制項，這裡 `dataGridView` 會顯示第二張資料表(`usersdata`)中所有客戶的資料，然後使用 SQL 語法 `COUNT(*)` 去計算總共有幾筆紀錄，這裡的更新鈕和刪除鈕設為不可選，使用 `Enabled=false`，並將顏色設為灰色，另外，我習慣使用 `try....catch` 包覆一段程式去執行，可以方便 Debug。



```
public void display() //顯示usersdata資料表中的所有紀錄
{
    try //try catch 方便Debug
    {
        using (SqlConnection cn = new SqlConnection())
        {
            cn.ConnectionString = cnstr;
            cn.Open();
            DataSet ds = new DataSet();
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM usersdata", cn);
            da.Fill(ds, "usersdata");
            dataGridView1.DataSource = ds.Tables[0];
            SqlCommand cmdCount = new SqlCommand("SELECT COUNT(*) FROM usersdata", cn); //計算usersdata資料表中有幾筆紀錄
            label9.Text = "全部的資料共" + cmdCount.ExecuteScalar().ToString() + "筆記錄"; //輸出計算的紀錄筆數
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}

private void button1_Click(object sender, EventArgs e) //全部客戶button，這裡只能唯讀沒辦法編輯
{
    display(); //顯示
    groupBox1.Visible = false; //將查詢客戶的groupBox1查詢框隱藏
    groupBox2.Visible = true; //將全部客戶(客戶編修)的groupBox2資料框顯示
    button6.Enabled = false; //更新欄設為不可選
    button7.Enabled = false; //刪除欄設為不可選
    button6.BackColor = Color.Gray; //更新欄設為灰色
    button7.BackColor = Color.Gray; //刪除欄設為灰色
}
```


點擊 dataGridView 的資料列兩下，可以將資料對應輸出到各自的 textBox，類似於 DataBinding，只是我是使用 **CellDoubleClick** 屬性，差別在於單擊或雙擊，另外，也會顯示每個客戶的照片，照片和客戶姓名相對應。

```
private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e) //CellDoubleClick屬性：當用戶雙擊單元格中的任意位置時發生。類似於DataBinding
{
    textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第一個欄位(客戶編號)的值
    textBox4.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第二個欄位(姓名)的值
    textBox5.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第三個欄位(身分證字號)的值
    textBox6.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第四個欄位(性別)的值
    textBox7.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第五個欄位(生日)的值
    textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第六個欄位(婚姻狀況)的值
    comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString(); //抓取dataGridView1顯示的usersdata資料表中滑鼠雙擊的那一列中第七個欄位(投保種類)的值
    pictureBox2.Image = new Bitmap("個人照/" + textBox4.Text + ".jpg"); //顯示個人照，將姓名對應到照片名稱
}
```

如果要進行更新或刪除就要到客戶編修這裡，**Enabled=true**。

```
private void button2_Click(object sender, EventArgs e) //客戶編修button
{
    groupBox1.Visible = false; //將查詢客戶的groupBox1查詢框隱藏
    groupBox2.Visible = true; //將全部客戶(客戶編修)的groupBox2資料框顯示
    button6.Enabled = true; //更新欄設為可選
    button7.Enabled = true; //刪除欄設為可選
    button6.BackColor = Color.Pink; //更新欄設為粉紅色
    button7.BackColor = Color.Pink; //刪除欄設為粉紅色
    display(); //顯示
}
```

投保種類我用 **comboBox 下拉式選單** 來選擇，將**第三張資料表**中(**typedata**)的險種欄位用 **while** 依序加到 comboBox 項目中。

T005	張學勳	I228765224	女
T006	甄紫單	O201789250	女
T007	張位件	L182648211	男

```
using (SqlConnection cn = new SqlConnection())
{
    cn.ConnectionString = cnstr;
    cn.Open();
    SqlCommand cmd2 = new SqlCommand("Select * from typedata", cn);
    SqlDataReader dr = cmd2.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr[0].ToString()); //將typedata資料表中的投保種類欄位依序加到全部客戶(客戶編修)的comboBox1項目中
        comboBox2.Items.Add(dr[0].ToString()); //將typedata資料表中的投保種類欄位依序加到查詢客戶的comboBox2項目中
    }
    dr.Close();
}
```

修改資料。



```
private void button6_Click(object sender, EventArgs e) //更新
{
    try //try catch 方便Debug
    {
        using (SqlConnection cn = new SqlConnection())
        {
            cn.ConnectionString = cnstr;
            cn.Open();
            SqlCommand cmd = new SqlCommand($"UPDATE usersdata SET 婚姻狀況=N'{textBox3.Text}',姓名=N'{textBox4.Text}',
            身分證字號=N'{textBox5.Text}',性別=N'{textBox6.Text}',生日=N'{textBox7.Text}',投保種類=N'{comboBox1.Text}'
            WHERE 客戶編號 = '{textBox2.Text}'", cn); //更新指令
            cmd.ExecuteNonQuery();
            MessageBox.Show("更新成功"); //輸出更新成功視窗
            display(); //dataGridView1跟著一起更新
            clear(); //清空所有textBox的值
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}
```

刪除資料，我有做一個刪除的確認鈕 `MessageBoxButtons.YesNo`，這樣就不會不小心誤刪，如果按是，則刪除成功，如果按否，則返回 `return`。



```
private void button7_Click(object sender, EventArgs e) //刪除
{
    try //try catch 方便Debug
    {
        using (SqlConnection cn = new SqlConnection())
        {
            if (MessageBox.Show("確定要刪除嗎?", "刪除", MessageBoxButtons.YesNo) == DialogResult.Yes) //刪除確認鈕，如果按是，則執行
            {
                cn.ConnectionString = cnstr;
                cn.Open();
                SqlCommand cmd = new SqlCommand($"DELETE FROM usersdata WHERE 客戶編號 = '{textBox2.Text}'", cn); //刪除指令
                cmd.ExecuteNonQuery();
                MessageBox.Show("刪除成功"); //輸出刪除成功視窗
                display(); //dataGridView1跟著一起更新
                clear(); //清空所有textBox的值
            }
            else //如果按否，則返回
            {
                return;
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}
```

刪除或更新完之後清空所有 textBox 的值。

search

全部客戶

查詢客戶

客戶維修

列印

資料框

客戶編號 婚姻狀況 2021年12月29日

姓名 身分證字號

性別 生日

投保種類

客戶編號	姓名	身分證字號	性別
T001	周閻伐	P124468549	男
T002	劉得濱	S154889067	男
T003	李曉隆	R275069243	女
T004	林政塵	T100459865	男
T005	張學勤	I228765224	女
T006	甄紫單	O201789250	女
T007	張位件	L182648211	男

更新

刪除

全部的資料共17筆記錄

```

public void clear() //清除textBox各欄位中的資料
{
    textBox2.Text = ""; //清空
    textBox3.Text = ""; //清空
    textBox4.Text = ""; //清空
    textBox5.Text = ""; //清空
    textBox6.Text = ""; //清空
    textBox7.Text = ""; //清空
    comboBox1.Text = ""; //清空
}

```

點擊查詢客戶，隱藏剛剛的資料框 `Visible=false`，顯示查詢框 `Visible=true`。

search

全部客戶

查詢客戶

客戶維修

列印

查詢框

請輸入想查詢的客戶姓名:

請輸入想查詢的投保種類:

```
private void button5_Click(object sender, EventArgs e) //查詢客戶button
{
    groupBox2.Visible = false; //將全部客戶(客戶編修)的groupBox2資料框隱藏
    groupBox1.Visible = true; //將查詢客戶的groupBox1查詢框顯示
}
```

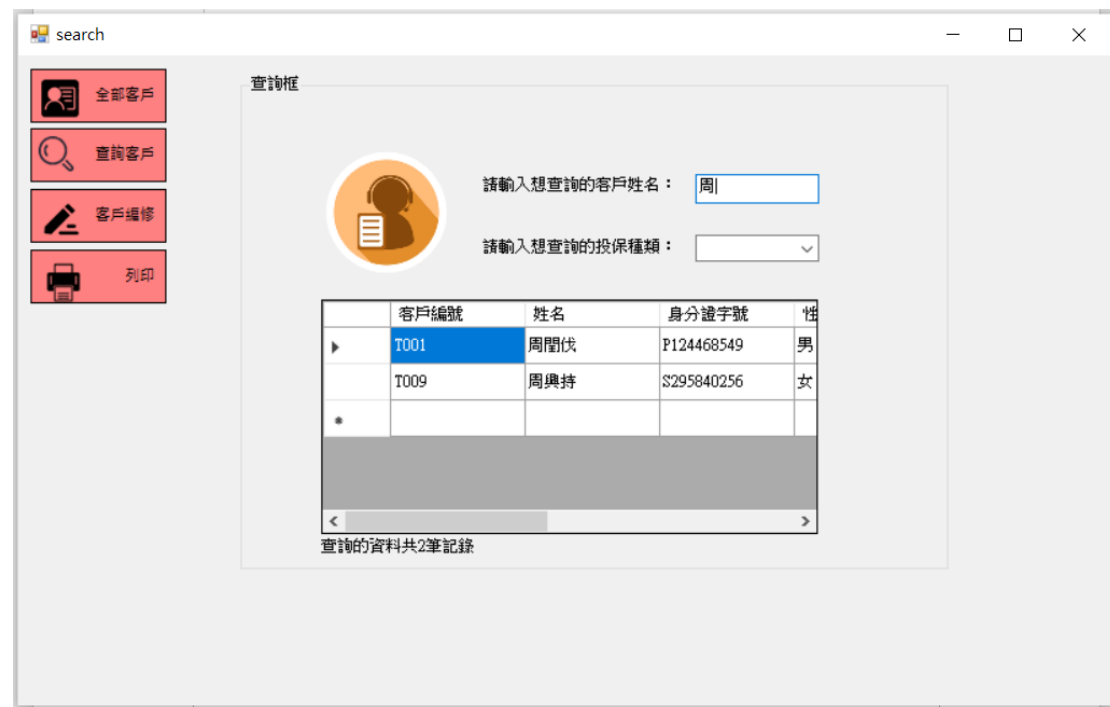
剛剛已經將第三張資料表中(typedata)的險種欄位用 while 依序加到 comboBox 項目中，現在就可以使用下拉式選單來做查詢，選擇汽機車強制險，就會顯示第二張資料表(usersdata)所有投保汽機車強制險的客戶，並顯示有幾筆紀錄 COUNT(*)。



```
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e) //下拉式選單發生改變
{
    using (SqlConnection cn = new SqlConnection())
    {
        cn.ConnectionString = cnstr;
        cn.Open();
        SqlDataAdapter adpt1 = new SqlDataAdapter($"SELECT * FROM usersdata WHERE 投保種類 =N'{comboBox2.Text}'", cn);
        //查詢comboBox2中的值
        DataSet ds2 = new DataSet();
        adpt1.Fill(ds2);
        dataGridView2.DataSource = ds2.Tables[0];
        label10.Visible = true; //原本label10被隱藏，現在將label10顯示
        SqlCommand cmdCount = new SqlCommand($"SELECT COUNT(*) FROM usersdata WHERE 投保種類 =N'{comboBox2.Text}'", cn);
        //計算usersdata資料表中，comboBox2下拉選單所選的投保種類，有幾筆紀錄
        label10.Text = "查詢的資料共" + cmdCount.ExecuteScalar().ToString() + "筆記錄";
        //輸出comboBox2下拉選單所選的投保種類有幾筆紀錄
    }
}
```

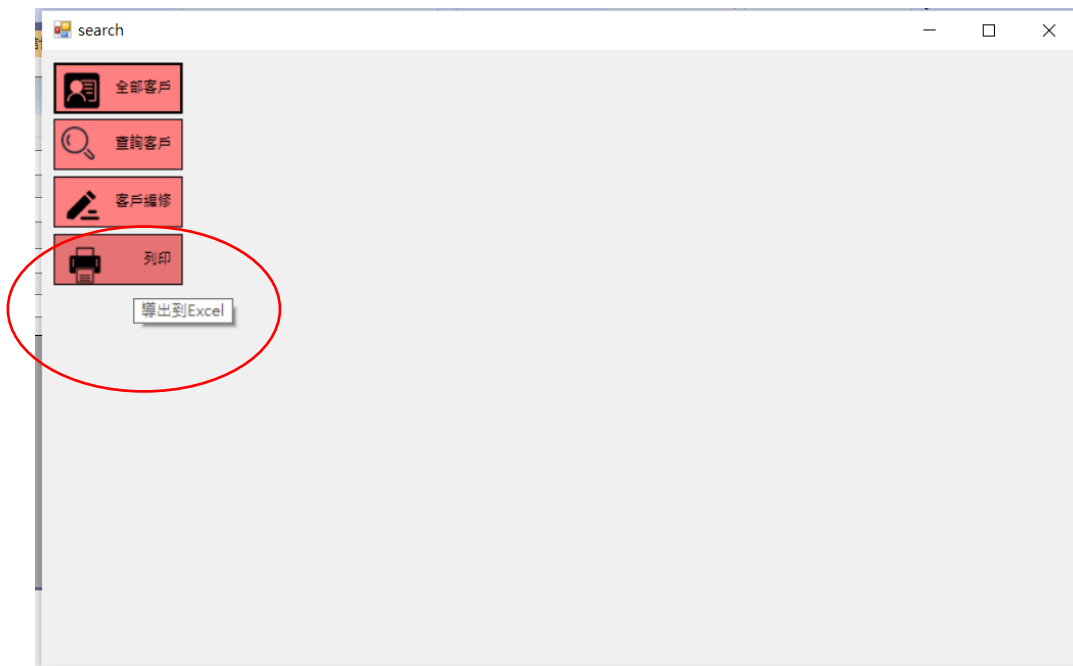
也可以使用姓名來做查詢，comboBox 的部分會自動清空，然後不需整個姓名都打出來，只要打出其中一個字就可以，不論是放在第一還是第二還是最後一個字都可以，因為我在前後都加了%，WHERE 姓名 LIKE ' %' +N' {textBox1.Text}' + ' %'，%是零或多個字元，記得使用 SQL 指令要加 N，不然資料會變成亂碼或問號，接著使用 TextChanged 屬性，只要欄位一

改變就會顯示不同資料，這樣就不用再做一個 Button 去呼叫它。



```
private void textBox1_TextChanged(object sender, EventArgs e)
//查尋客戶中的姓名查詢textBox，TextChanged屬性是只要欄位一改變就顯示不同資料
{
    try //try catch 方便Debug
    {
        using (SqlConnection cn = new SqlConnection())
        {
            cn.ConnectionString = cnstr;
            cn.Open();
            SqlDataAdapter adpt = new SqlDataAdapter($"SELECT * FROM usersdata WHERE 姓名 LIKE '%'+ N'{textBox1.Text}' + '%'", cn);
            //查詢textBox1輸入的姓名，%是零或多個字元，前面要加N
            DataSet dsl = new DataSet();
            adpt.Fill(dsl);
            dataGridView2.DataSource = dsl.Tables[0];
            label10.Visible = true; //原本label10被隱藏，現在將label10顯示
            SqlCommand cmdCount = new SqlCommand($"SELECT COUNT(*) FROM usersdata WHERE 姓名 LIKE '%'+ N'{textBox1.Text}' + '%'", cn);
            //計算usersdata資料表中，textBox1輸入的姓名，有幾筆紀錄
            label10.Text = "查詢的資料共" + cmdCount.ExecuteScalar().ToString() + "筆記錄"; //輸出textBox1輸入的姓名有幾筆紀錄
            comboBox2.Text = ""; //將投保種類comboBox2清空
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}
```

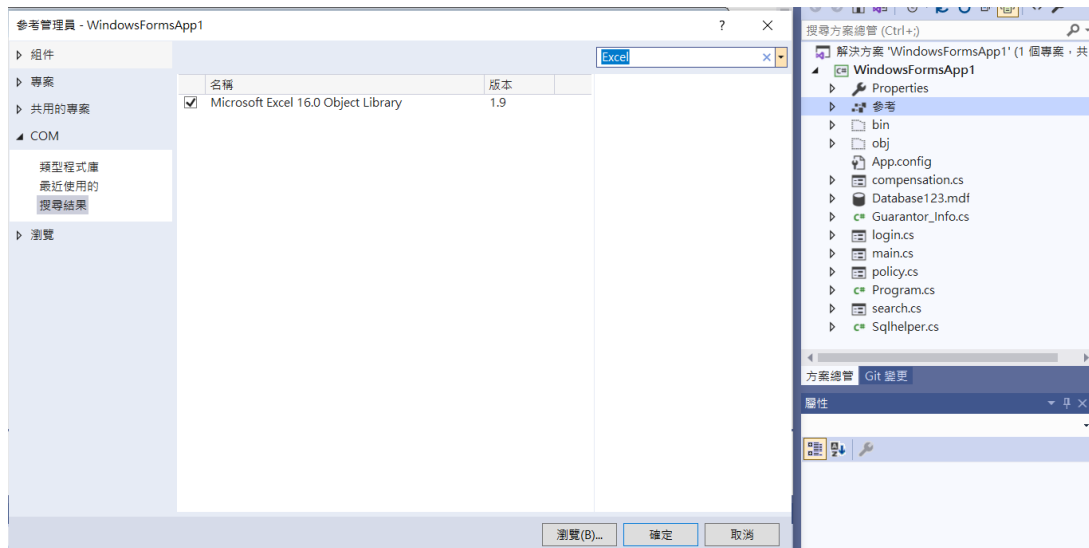
再來列印的部分，我是將資料導出到 Excel，再到 Excel 做列印，將滑鼠移到列印按鈕的地方，會顯示”導出到 Excel 的字串”，那是使用 **ToolTip 控制項**。接著點擊列印，就會將**第二張**資料表(usersdata)的所有客戶導出到 Excel。



Excel screenshot showing a table with columns: 客戶編號 (Customer ID), 姓名 (Name), 身分證字號 (ID Number), 性別 (Gender), 生日 (Date of Birth), 婚姻狀況 (Marital Status), 投保種類 (Insurance Type).

客戶編號	姓名	身分證字號	性別	生日	婚姻狀況	投保種類
T001	周開茂	P1244685	男	1966/5/6	已婚	醫療險
T002	劉得清	S1548890	男	1971/9/12	已婚	壽險
T003	李鴻隆	R2750692	女	1984/12/4	未婚	醫療險
T004	林政隆	T1004598	男	1975/1/2	未婚	失能險
T005	張學鵬	I2876522	女	1991/6/30	已婚	旅遊平安險
T006	鄭紫華	O2017892	女	1989/7/19	未婚	超額責任險
T007	張位件	L1826482	男	1977/8/23	已婚	癌症險
T008	鄭紹郎	N2584265	女	1969/4/1	未婚	長期照顧險
T009	周興持	S2958402	女	1990/8/8	未婚	意外險
T010	城達	I24860305	女	1983/9/16	已婚	傷害險
T011	古源拉	P1486305	男	1988/7/11	未婚	生存險
T012	吳夢蓉	I26581305	女	1968/2/22	已婚	汽機車強制險
T013	吳其鳳	T2052036	女	1972/3/29	已婚	汽機車強制險
T014	劉得清	S1548890	男	1971/9/12	已婚	醫療險
T015	鄭紹郎	N2584265	女	1969/4/1	未婚	癌症險
T016	楊天勝	A1234567	男	2001/1/6	未婚	醫療險
T017	高昇	B2234567	女	2001/8/9	未婚	汽機車強制險

首先，要到加入參考的地方搜尋並勾選 Microsoft Excel 16.0 Object Library，然後到程式碼的地方 using 這段 using Microsoft.Office.Interop.Excel;，並且將陣列使用 for 迴圈導出標題列，再用雙層 for 迴圈導出資料即可。



```
private void button3_Click(object sender, EventArgs e) //導出資料到Excel
{
    display(); //將usersdata資料表呈現在dataGridView1
    try //try catch 方便Debug
    {
        Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();
        Workbook wb = excel.Workbooks.Add(XlSheetType.xlWorksheet);
        Worksheet ws = (Worksheet)excel.ActiveSheet;
        excel.Visible = true;
        for (int i = 1; i < dataGridView1.Columns.Count + 1; i++) //導出標題列
        {
            ws.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
        }
        for (int i = 0; i < dataGridView1.Rows.Count; i++) //導出資料
        {
            for (int j = 0; j < dataGridView1.Columns.Count; j++)
            {
                if (dataGridView1.Rows[i].Cells[j].Value != null && dataGridView1.Rows[i].Cells[j].Value.ToString() != "")
                {
                    ws.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
                }
                else
                {
                    ws.Cells[i + 2, j + 1] = "";
                }
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}
```

- **第四部分(compensation.cs)**

點擊理賠按鈕顯示這個畫面(compensation.cs)。上面有一排不同的險種是使用 **tabControl 控制項**，可以分別放入各自需要的資料，上面 dataGridView 是**第四張**資料表(cimpensationdata)的理賠資料，下面比較小的 dataGridView 是**第五張**資料表(money)的公司餘額。

compensation

傷害險 壽險 失能險 意外險 旅遊平安險 汽機車強制險 生存險 癌症險 超額責任險 醫療險

	客戶編號	姓名	投保種類	金額
▶	T010	城隆	傷害險	20000
•				

搜尋理賠金額大於等於 的紀錄 客戶編號:

	餘額
▶	99780000

公司餘額:
理賠金額:

```
SqlCommand cmd13 = new SqlCommand("SELECT * FROM money", cn); //money資料表的餘額
SqlDataAdapter sqldata = new SqlDataAdapter(cmd13);
DataSet dataset = new DataSet();
sqldata.Fill(dataset);
dataGridView13.DataSource = dataset.Tables[0];
```

```
SqlCommand cmd = new SqlCommand("SELECT * FROM compensationdata WHERE 投保種類='傷害險'", cn); //tabControl1傷害險的部分
SqlDataAdapter da = new SqlDataAdapter(cmd);
DataSet ds = new DataSet();
da.Fill(ds);
dataGridView1.DataSource = ds.Tables[0];
```

然後雙擊上面的其中一筆紀錄，label 會顯示對應的客戶編號和理賠金額。

compensation

傷害險 壽險 失能險 意外險 旅遊平安險 汽機車強制險 生存險 癌症險 超額責任險 醫療險

	客戶編號	姓名	投保種類	金額
▶	T010	城隆	傷害險	20000
•				

搜尋理賠金額大於等於 的紀錄 客戶編號: T010

	餘額
▶	99780000

公司餘額:
理賠金額: 20000

```
private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e) //傷害險 CellDoubleClick屬性：當用戶雙擊單元格中的任意位置時發生。類似於DataBinding
{
    label8.Visible = true;
    label8.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString(); //抓取dataGridView1顯示的compensationdata資料表中滑鼠雙擊的那一列中第一個欄位(客戶編號)的值
    label12.Visible = true;
    label12.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString(); //抓取dataGridView1顯示的compensationdata資料表中滑鼠雙擊的那一列中第三個欄位(金額)的值
}
```

然後雙擊下面的餘額，label 會顯示對應的公司餘額。

```
private void dataGridView13_CellDoubleClick(object sender, DataGridViewCellEventArgs e) //公司餘額 CellDoubleClick屬性：當用戶雙擊單元格中的任意位置時發生。類似於DataBinding
{
    label6.Visible = true;
    label6.Text = dataGridView13.Rows[e.RowIndex].Cells[0].Value.ToString(); //抓取dataGridView13顯示的money資料表中滑鼠雙擊的那一列中第一個欄位(餘額)的值
}
```

這時候就可以按理賠，它會問我們確定要理賠嗎，按是之後，顯示理賠成功以及公司餘額還有多少，然後該筆紀錄就會從第四張資料表(compensationdata)中刪除，第五張資料表(money)餘額也會跟著變動，並將 label 清空。

```

using (SqlConnection cn = new SqlConnection())
{
    if (MessageBox.Show("確定要理賠嗎?", "理賠", MessageBoxButtons.YesNo) == DialogResult.Yes) //理賠確認鈕，如果按是，則執行
    {
        cn.ConnectionString = cnstr;
        cn.Open();

        SqlCommand scmd = new SqlCommand($"DELETE FROM compensationdata WHERE 客戶編號 = '{label8.Text}'", cn);
        //理賠之後，compensationdata資料表中刪除該筆紀錄
        scmd.ExecuteNonQuery();
        DataSet ds12 = new DataSet(); //dataGridView12跟著更新
        SqlDataAdapter da12 = new SqlDataAdapter("SELECT * FROM compensationdata", cn);
        da12.Fill(ds12, "compensationdata");
        dataGridView12.DataSource = ds12.Tables[0];

        int money = int.Parse(label12.Text); //取得compensationdata資料表中所選的理賠資料金額
        int total = int.Parse(label6.Text); //取得money資料表中的公司餘額
        int sum = total - money; //設一個變數計算理賠後新的公司餘額
        MessageBox.Show($"理賠成功，公司餘額剩餘(sum)"); //輸出理賠成功，並顯示公司剩餘金額有多少
        label6.Text = ""; //將label6顯示的公司餘額清空
        label8.Text = ""; //將label8顯示的客戶編號清空
        label12.Text = ""; //將label12顯示的理賠金額清空

        SqlCommand scmd1 = new SqlCommand($"UPDATE money SET 餘額={sum}", cn); //理賠之後，money資料表中公司餘額跟著變動
        scmd1.ExecuteNonQuery();
        DataSet ds13 = new DataSet(); //dataGridView13跟著更新
        SqlDataAdapter da13 = new SqlDataAdapter("SELECT * FROM money", cn);
        da13.Fill(ds13, "money");
        dataGridView13.DataSource = ds13.Tables[0];
        display();
    }
    else //如果按否，則返回
    {
        return;
    }
}

```

我在 tabControl 控制項的最後一格全部這裡，用 LINQ 方法寫了一個可以搜尋第四張資料表(compensationdata)中金額大於等於多少的紀錄，然後將金額由大排到小，在這裡也可以做理賠的動作。

compensation

意外險 旅遊平安險 汽機車強制險 生存險 癌症險 超額責任險 醫療險 長期照顧險 全部

客戶編號	姓名	投保種類	金額
T008	鄭紹邱	長期照顧險	300000
T004	林政塵	失能險	200000
T016	楊天勝	醫療險	150000
T007	張位件	癌症險	70000
T015	鄭紹邱	癌症險	70000
T011	古添埕	生存險	60000
T009	周興持	意外險	50000

搜尋理賠金額大於等於 50000 的紀錄 搜尋 客戶編號: 計算機 理賠

餘額
99760000

公司餘額: 理賠金額:

```

private void button3_Click(object sender, EventArgs e) //LINQ方法搜尋金額
{
    try //try catch 方便Debug
    {
        using (SqlConnection cn = new SqlConnection())
        {
            cn.ConnectionString = cnstr;
            cn.Open();

            SqlDataAdapter daE = new SqlDataAdapter("SELECT * FROM compensationdata ORDER BY 金額 DESC", cn);
            DataSet dts = new DataSet();
            daE.Fill(dts, "compensationdata");
            dataGridView12.DataSource = dts.Tables["compensationdata"];
            DataTable dtEmp = dts.Tables["compensationdata"];
            var emp = from p in dtEmp.AsEnumerable()
                where p.Field<int>("金額") >= int.Parse(textBox1.Text) //查詢compensationdata資料表中金額>=輸入的金額的資料
                orderby p.Field<int>("金額") descending //金額由大到小
                select new
                {
                    客戶編號 = p.Field<string>("客戶編號"),
                    姓名 = p.Field<string>("姓名"),
                    投保種類 = p.Field<string>("投保種類"),
                    金額 = p.Field<int>("金額")
                };
            dataGridView12.DataSource = emp.ToList();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message); //輸出錯誤訊息
    }
}

```

點擊計算機按鈕可以呼叫 Windows form 內建的計算機功能。



```

private void button1_Click(object sender, EventArgs e) //計算機按鈕
{
    System.Diagnostics.Process.Start("C:\\WINDOWS\\system32\\calc.exe"); //Windows form內建的功能
}

```

- 第五部分(policy.cs)

新增表單的 Form – policy

policy 表單的主要功能為，供使用者輸入資料，至資料庫的資料表中。

```
private void policy_Load(object sender, EventArgs e)
{
    // TODO: 這行程式碼會將資料載入 'database123DataSet.usersdata' 資料表。您可以視需要進行移動或移除。
    this.usersdataTableAdapter.Fill(this.database123DataSet.usersdata);
    //要保人性別ComboBox選項
    cmb_gender.Items.Add("女");
    cmb_gender.Items.Add("男");
    //要保人婚姻狀況ComboBox選項
    cmb_marriage.Items.Add("未婚");
    cmb_marriage.Items.Add("已婚");
    //要保人投保種類ComboBox選項
    cmb_type.Items.Add("傷害險");
    cmb_type.Items.Add("壽險");
    cmb_type.Items.Add("失能險");
    cmb_type.Items.Add("意外險");
    cmb_type.Items.Add("旅遊平安險");
    cmb_type.Items.Add("汽機車強制險");
    cmb_type.Items.Add("生存險");
    cmb_type.Items.Add("癌症險");
    cmb_type.Items.Add("超額責任險");
    cmb_type.Items.Add("醫療險");
    cmb_type.Items.Add("長期照顧險");
}
```

1 個參考

在 `policy_Load` 方法裡，主要寫入三個下拉式選單選項分別為性別、婚姻狀況、投保類型。

主要功能介紹

1. 範例按鈕 — 載入預設資料以供使用者參考

點擊範例按鈕後，將會把姓名、未婚、姓名、投保種類、生日、身份證字號預設資料顯示。

執行畫面

要保人

客戶編號

姓名: 測試員 婚姻狀況: 未婚

性別: 女 投保種類: 旅遊平安險

生日: 1987年 2月 3日 身分證字號: S223289982

客戶編號	姓名	身分證字號	性別	生日
T001	周閏伐	P224468549	男	1986/5/6
T002	劉得滑	S554889067	男	1971/9/12
T003	李曉隆	R875069243	女	1984/12/14

範例 匯入 確認 清除 離開

程式碼

```
private void btnload_Click(object sender, EventArgs e)
{
    txt_name.Text = "測試員";
    cmb_gender.SelectedIndex = 0;
    dt_br.Value = DateTime.Parse("1987/2/3");
    txt_id.Text = "S223289982";
    cmb_marriage.SelectedIndex = 0;
    cmb_type.SelectedIndex = 4;
}
```

2. 匯入按鈕

點選匯入按鈕後，出現 insert 介面。

執行畫面

policy

要保人

客戶編號

姓名 婚姻狀況

性別 投保種類

生日 2021年12月29日 身分證字號

	客戶編號	姓名	身分證字號	性別	生日
▶	T001	周閏伐	P224468549	男	1986/5/6
	T002	劉得滑	S554889067	男	1971/9/12
	T003	李曉隆	R875069243	女	1984/12/14

範例
匯入
確認
清除
離開

程式碼

```
private void btn_insert_Click(object sender, EventArgs e)
{
    insert insert = new insert(); //顯示insert Form介面
    insert.Show();
}
```

3. 確認按鈕

按下確認後，將輸入好的資料匯入資料庫(usersdata 資料表)裡
執行畫面
(無誤時)

policy

要保人

客戶編號 T021

姓名 測試員 婚姻狀況 未婚

性別 女 投保種類 旅遊平安險

生日 1987年 2月 3日 身分證字號 S223289982

範例

匯入

確認

清除

離開

已輸入資料

確定

(有資料未填時)

policy

要保人

客戶編號

姓名 婚姻狀況

性別 投保種類

生日 2021年12月29日 身分證字號

範例

匯入

確認

清除

離開

客戶編號	姓名	身分證字號
T001	周閏伐	P224468549
T002	劉得滑	S554889067
T003	李曉隆	R875069243

資料空白，無法存入用戶資料

確定

判斷輸入的值，若有為空，跳出對話框。

(身分證錯誤時)

要保人

客戶編號

姓名: 測試員 婚姻狀況: 未婚

性別: 女 投保種類: 旅遊平安險

生日: 1987年 2月 3日 身分證字號: S223

	客戶編號	姓名	身分證字號
▶	T001	周閏伐	P224468549
	T002	劉得滑	S554889067
<	T003	李曉隆	R875069243

身分證長度錯誤

確定

範例 匯入 確認 清除 離開

程式碼

(此程式碼為確認輸入的資料是否有空值的輸入)

```
private void btnCFM_Click(object sender, EventArgs e)
{
    //判斷是否有資料為空值
    if (string.IsNullOrEmpty(txt_name.Text)
        || string.IsNullOrEmpty(txt_id.Text)
        || string.IsNullOrEmpty(cmb_type.Text)
        || string.IsNullOrEmpty(cmb_gender.Text)
        || string.IsNullOrEmpty(cmb_marriage.Text))
    {
        MessageBox.Show("資料空白，無法存入用戶資料");
    }
}
```

(此程式碼，包含驗證身分證長度、建立物件、計算客戶編號方法，使用 Sqlhelper.Insertgrt 方法、view 顯示方法，以下分別介紹)

```
else
{
    if (txt_id.Text.Length == 10)
    {
        Guarantor_Info addgrt = new Guarantor_Info(); //建立addgrt物件

        //int i = 20;
        //addgrt.Client_ID = txt_clientid.Text; 自己輸入客戶編號
        //label9.Text = "T" + i.ToString("000");
        //i++;
        countid();
        addgrt.Client_ID = label9.Text; //將顯示的要保人客戶編號存入addgrt.Client_ID裡
        addgrt.Name = txt_name.Text; //將輸入的要保人名字存入addgrt.Name裡
        addgrt.ID = txt_id.Text; //將輸入的要保人身分證字號存入addgrt.ID裡
        addgrt.Birth = dt_br.Value.ToShortDateString(); //將生日DateTimePicker的值使用ToShortDateString捨去時間存入addgrt.Birth裡
        addgrt.Marriage = cmb_marriage.Text; //將婚姻狀況ComboBox選擇顯示的文字加入到addgrt
        addgrt.Gender = cmb_gender.Text; //將性別ComboBox選擇顯示的文字加入到addgrt.Marriage裡
        addgrt.Type = cmb_type.Text; //將投保種類ComboBox選擇顯示的文字加入到addgrt.Type裡
        sqlServerSingleton.Insertgrt(addgrt);
        view(); //顯示資料方法
    }
    else
    {
        MessageBox.Show("身分證長度錯誤");
    }
}
```

首先是 `countid()` 方法，其功能為計算目前客戶編號

```
1 10參考
private void countid()
{
    SqlConnection con = new SqlConnection(connectionString);
    con.Open();
    SqlCommand cmd = new SqlCommand("SELECT count(客戶編號) FROM usersdata", con);
    int i = Convert.ToInt32(cmd.ExecuteScalar());
    con.Close();
    i++;
    label9.Text = "T" + i.ToString("000");
}
```

使用 `count(客戶編號)` 可以找出目前資料表中的客戶編號數，若採用傳統式的計數器(ex : `int i = 20; label9.Text = "T" + i.ToString("000"); i++;`)，在專案結束後，下次執行專案時資料會重複輸入。

```
Guarantor_Info addgrt = new Guarantor_Info(); //建立addgrt物件
```

Database Explorer
c# Guarantor_Info.cs

一個類別，以供資料表各欄位進行自動實作。

```
namespace WindowsFormsApp1
{
    3 個參考
    class Guarantor_Info
    {
        //設定自動實作屬性

        2 個參考
        public string Client_ID { get; set; }
        2 個參考
        public string Name { get; set; }
        2 個參考
        public string ID { get; set; }
        2 個參考
        public string Gender { get; set; }
        2 個參考
        public string Birth { get; set; }
        2 個參考
        public string Marriage { get; set; }
        2 個參考
        public string Type { get; set; }
    }
}

addgrt.Client_ID = label9.Text; //將顯示的要保人客戶編號存入addgrt.Client_ID裡
addgrt.Name = txt_name.Text; //將輸入的要保人名字存入addgrt.Name裡
addgrt.ID = txt_id.Text; //將輸入的要保人身分證字號存入addgrt.ID裡
addgrt.Birth = dt_br.Value.ToShortDateString(); //將生日DateTimePicker的值使用ToShortDateString捨去時間存入addgrt.Birth裡
addgrt.Marriage = cmb_marriage.Text; //將婚姻狀況ComboBox選擇顯示的文字加入到addgrt
addgrt.Gender = cmb_gender.Text; //將性別ComboBox選擇顯示的文字加入到addgrt.Marriage裡
addgrt.Type = cmb_type.Text; //將投保種類ComboBox選擇顯示的文字加入到addgrt.Type裡
```

預先設定 `sqlServerSingleton` 為 `Sqlhelper.cs` 的物件並使用 `Sqlhelper.GetInstance` 方法

```
private Sqlhelper sqlServerSingleton = Sqlhelper.GetInstance();
```

```

    }
    // lock 關鍵字可以用來確保代碼塊完成運行
    1 個參考
    public static Sqlhelper GetInstance()
    {
        if (sqlServerSingleton == null)
        {
            lock (locker)
            {
                if (sqlServerSingleton == null)
                {
                    sqlServerSingleton = new Sqlhelper();
                }
            }
        }
        return sqlServerSingleton;
    }
}

```

然後在 btncfm 裡使用 `sqlServerSingleton.Insertgrt(addgrt);` 將資料存入 addgrt 物件

Insertgrt 程式碼(在 Sqlhelper 裡)

```

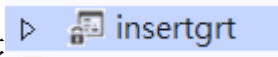
1 個參考
public void Insertgrt(Guarantor_Info addgrt)
{
    //開啟 SqlConnection 物件
    SqlConnection connection = new SqlConnection(connectionString);
    try
    {
        SqlCommand cmd = new SqlCommand();

        cmd.Connection = connection;
        connection.Open();
        cmd.CommandText = "[dbo].[insertgrt]"; //設定預存程序 insertgrt
        cmd.CommandType = CommandType.StoredProcedure; //呼叫預存程序
        cmd.Parameters.AddWithValue("@client_id", addgrt.Client_ID);
        cmd.Parameters.AddWithValue("@name", addgrt.Name);
        cmd.Parameters.AddWithValue("@id", addgrt.ID);
        cmd.Parameters.AddWithValue("@gender", addgrt.Gender);
        cmd.Parameters.AddWithValue("@birth", addgrt.Birth);
        cmd.Parameters.AddWithValue("@marriage", addgrt.Marriage);
        cmd.Parameters.AddWithValue("@type", addgrt.Type);

        int i = cmd.ExecuteNonQuery();
        if (i > 0) { MessageBox.Show("已輸入資料"); }
        connection.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.ToString());
    }
}

```

呼叫預存程序



```
新(U)
1 CREATE PROCEDURE [dbo].[insertgrt]
2     @client_id nvarchar(50),
3     @name nvarchar(50),
4     @id nvarchar(50),
5     @gender nvarchar(50),
6     @birth nvarchar(50),
7     @marriage nvarchar(50),
8     @type nvarchar(50)
9 AS
10 BEGIN
11     INSERT into usersdata(客戶編號,姓名,身分證字號,性別,生日,婚姻狀況,投保種類) values (@client_id,@name,@id,@gender,@birth,@marriage,@type)
12 END
13 RETURN 0
```

(將從使用者輸入的資料由 addgrt 物件自動存取的各欄位，進行 AddWithValue 後，將獲取到的資料存入 usersdata 資料表的欄位)

view() 方法程式碼，顯示 usersdata 資料表，以供確認是否將資料輸入。

```
public void view()
{
    SqlConnection con = new SqlConnection(connectionString);
    con.Open();
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM usersdata", con);
    da.Fill(ds, "usersdata");
    dataGridView1.DataSource = ds.Tables[0];
}

else
{
    MessageBox.Show("身分證長度錯誤");
}
}
```

如果 txt_id(身分證) 輸入的長度正確(10)就進行賦值，如果沒有則顯示對話框提示。

4. 清除按鈕

清除目前欄位裡的資料

policy

要保人

客戶編號

姓名 婚姻狀況

性別 投保種類

生日 身分證字號

客戶編號	姓名	身分證字號	性別	生日
T001	周閏伐	P224468549	男	1986/5/6
T002	劉得滑	S554889067	男	1971/9/12
T003	李曉隆	R875069243	女	1984/12/14

範例
匯入
確認
清除
離開

執行結果

要保人

客戶編號

姓名 婚姻狀況

性別 投保種類

生日 身分證字號

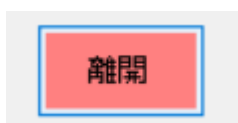
範例

程式碼

1 個參考

```
private void btnClear_Click(object sender, EventArgs e)
{
    //將表單清除
    //txt_clientid.Text = string.Empty;
    label19.Text = string.Empty;
    txt_name.Text = string.Empty;
    txt_id.Text = string.Empty;
    cmb_gender.SelectedIndex = -1; //設為0時，會是第一個選項(女)，設-1則為空值
    dt_br.Value = DateTime.Now; //時間設為現在
    cmb_marriage.SelectedIndex = -1;
    cmb_type.SelectedIndex = -1;
}
```

5.



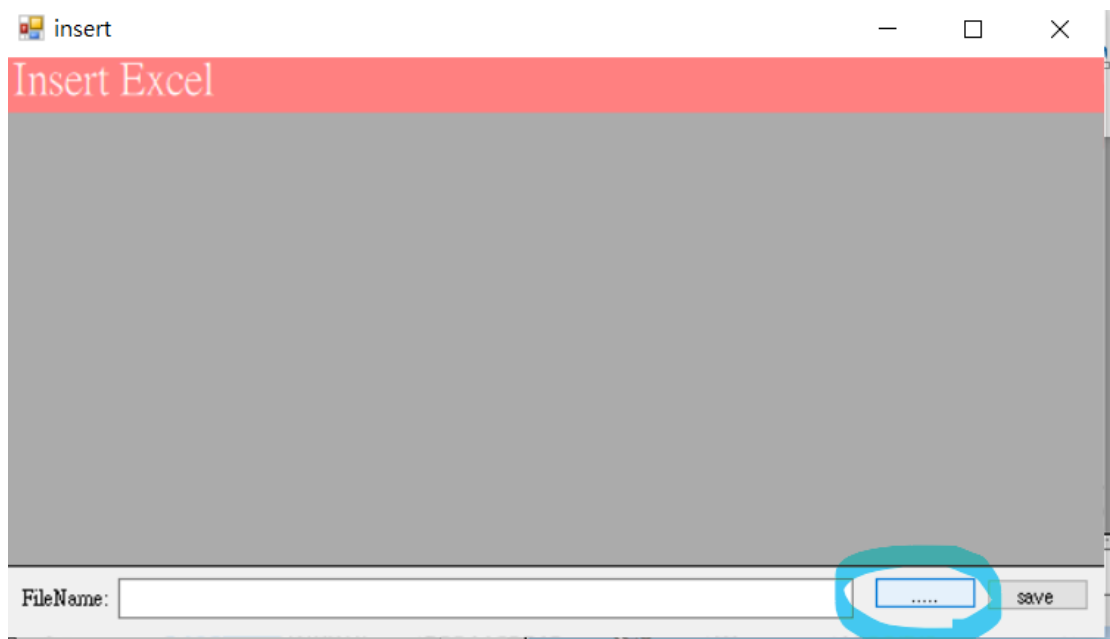
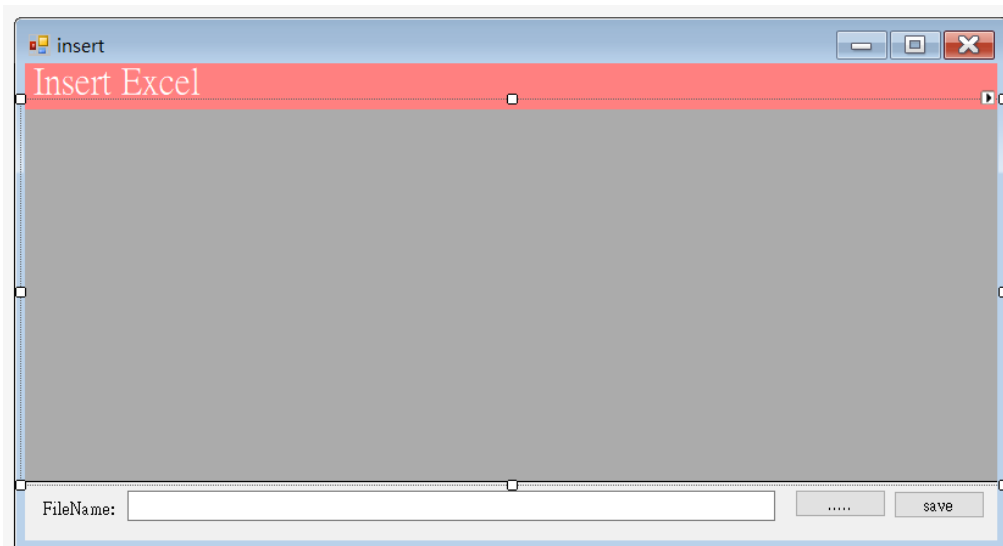
關閉視窗

1 個參考
`private void btn_end_Click(object sender, EventArgs e)`
{
 this.Close();
}

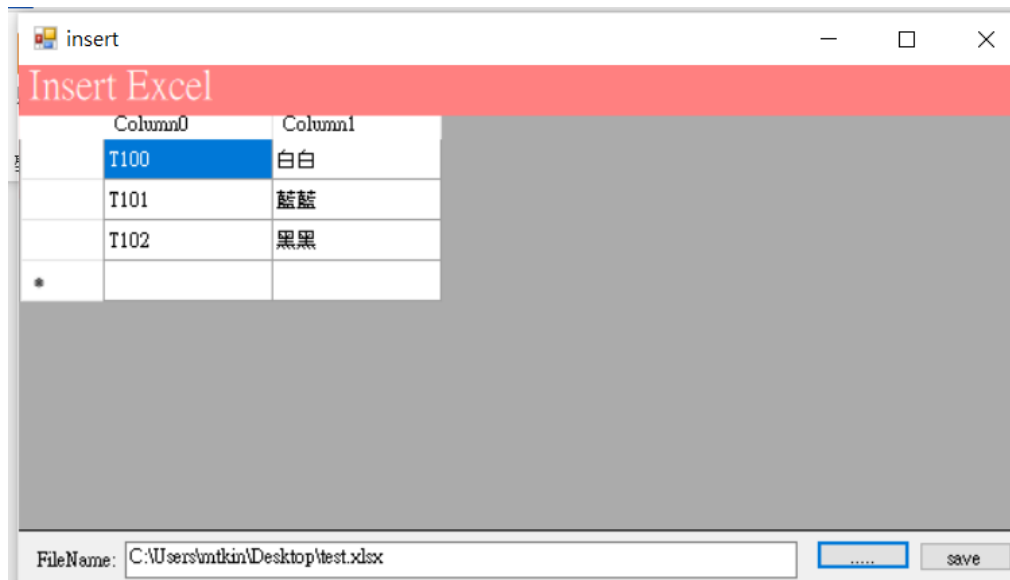
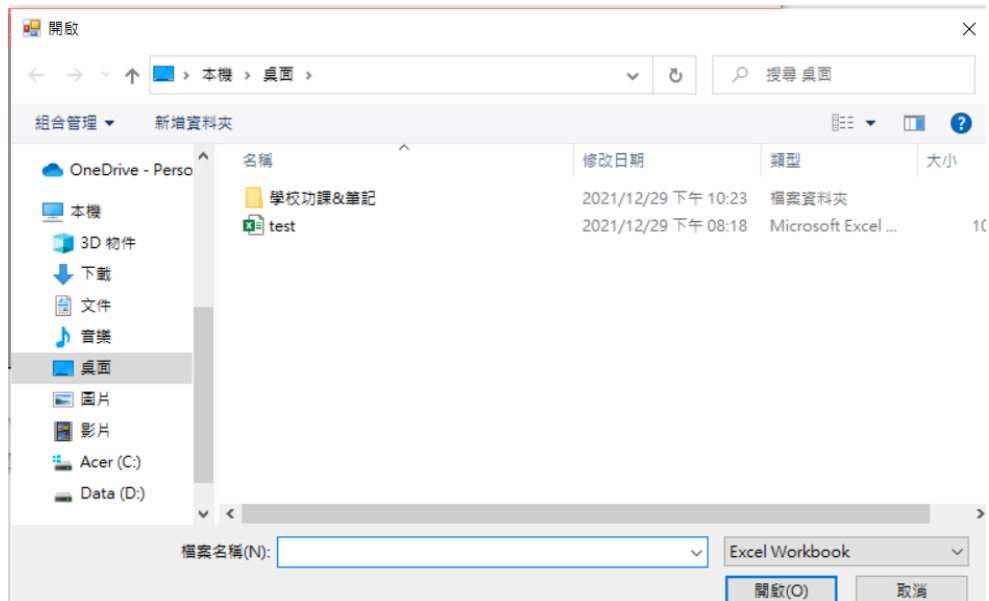
第六部分(insert.cs)

Insert Form

功能，將 Excel 資料抓取路徑，並存入資料庫



(使用 btnbro 抓取 excel 檔案資料，並同時存入 dataGridView1 裡)



```

1 個參考
private void btnBro_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog= new OpenFileDialog() { Filter = "Excel Workbook(*.xlsx)|Excel 97-2003 Workbook(*.xls) })// 檔案格式選取
    {
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            txtFile.Text = openFileDialog.FileName; //顯示路徑名稱
            var stream = File.Open(openFileDialog.FileName, FileMode.Open, FileAccess.Read); //開啟資料
            var reader = ExcelReaderFactory.CreateReader(stream); //讀取資料
            var result = reader.AsDataSet(); //將資料轉成DataSet存在result中
            var tables = result.Tables.Cast<DataTable>();

            foreach (DataTable table in tables)
            {
                dataGridView1.DataSource = table;
            }
        }
    }
}

```



```
//1.檔案位置
//2.提供者名稱 Microsoft.Jet.OLEDB.4.0適用於2003以前版本，Microsoft.ACE.OLEDB.12
string ProviderName = "Microsoft.ACE.OLEDB.12.0;";

//3.Excel版本，Excel 8.0 針對Excel2000及以上版本，Excel5.0 針對Excel97。
string ExtendedString = "'Excel 8.0;";

//4.第一行是否為標題(結尾區隔)
string HDR = "YES;";

//5.IMEX=1 通知驅動程序始終將「互混」數據列作為文本讀取(結尾區隔,'文字結尾)
string IMEX = "0;";

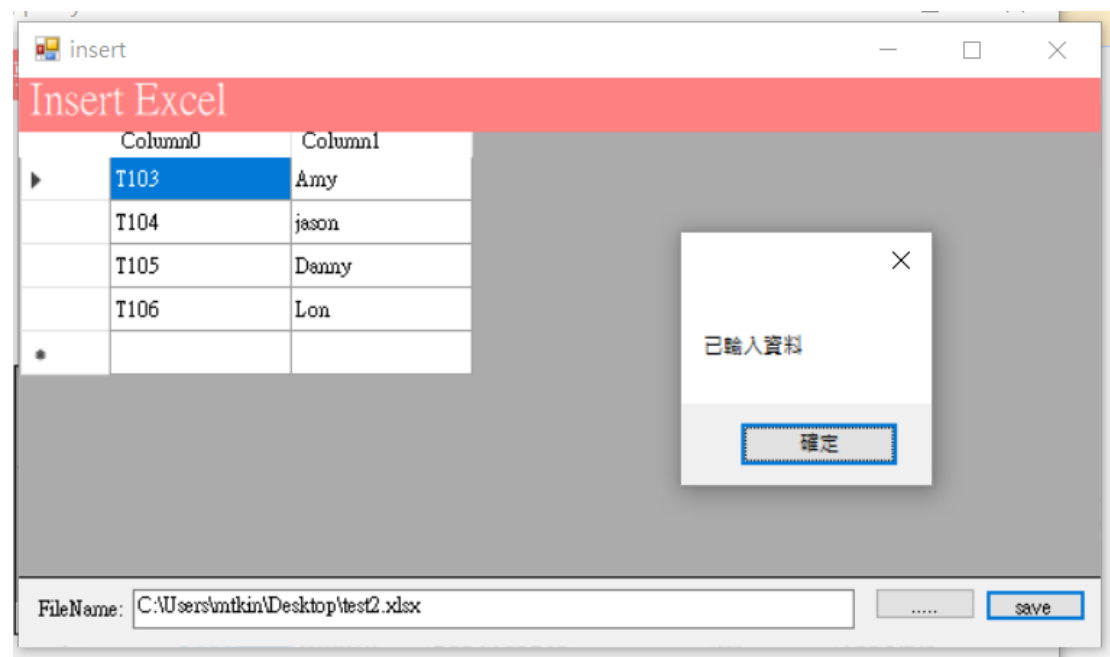
//連線字串
string conString =
    "Provider=" + ProviderName +
    "Data Source=" + FilePath + ";" +
    "Extended Properties=" + ExtendedString +
    "HDR=" + HDR +
    "IMEX=" + IMEX;

// string conString = @"Provider=Microsoft.ACE.OLEDB.12.0;"+ "Data Source = " + F

OleDbConnection con = new OleDbConnection(conString);
con.Open();
string queryString = "SELECT * FROM [" + SheetName + "$]";
DataSet ds = new DataSet();
OleDbDataAdapter dr = new OleDbDataAdapter(queryString, con);
dr.Fill(ds);
dataTable = ds.Tables[0];
this.dataGridView1.DataSource = dataTable;
con.Close();
```

(失敗案例)

錯誤訊息'Microsoft.ACE.OLEDB.12.0' 提供者並未登錄於本機電腦上



客戶編號	姓名
T103	Amy
T104	jason
T105	Danny
T106	Lon
NULL	NULL

1 個參考

```
private void btn_imp_Click(object sender, EventArgs e)
{
    con.Open();
    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        SqlCommand cmd = new SqlCommand("Insert into test(客戶編號,姓名) values ('"
            + dataGridView1.Rows[i].Cells[0].Value + "','" +
            dataGridView1.Rows[i].Cells[1].Value + "','" + "')", con);
        cmd.ExecuteNonQuery();
    }
    MessageBox.Show("已輸入資料");
    SqlDataAdapter da = new SqlDataAdapter("select * from test ", con);
    DataSet dt = new DataSet();
    da.Fill(dt);
    dataGridView1.DataSource = dt;
    con.Close();
}
```

(此功能為將資料存入資料庫，利用 btnBro 抓取到的 excel 呈現在 dataGridView 上，抓取 dataGridView 的資料行，進行資料存取。