

系统分析与设计课程设计实验报告

学号：

姓名：

指导老师：

完成时间：

# 实验五 界面设计与文件存取

## 一、实验目的

(1)练习 SQL Server 数据库的创建与使用方法。

(2)练习存储过程的创建与调用方法练习简单的统计査询方法。

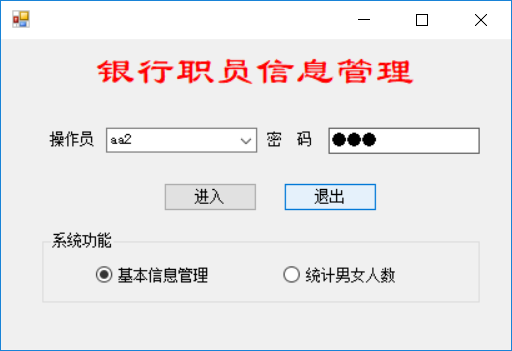
(3)练习绑定 DataGridView 到数据源的方法。

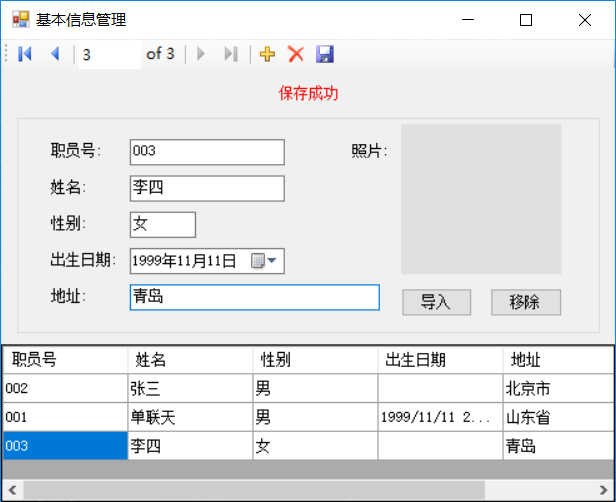
(4)练习绑定其他控件到数据源的方法。

(5)练习数据库中图片的显示和存取方法。。

## 二、实验结果

界面运行截图：





## 三、主要代码

## [FormMain.cs]

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Data.SqlTypes;

namespace BankEmployee

{

public partial class FormMain : Form

{

public FormMain()

{

InitializeComponent();

}

private void FormMain\_Load(object sender, EventArgs e)

{

this.selectAllUserNameFromOperatorTableAdapter.Fill(this.employeeDataSet1.SelectAllUserNameFromOperator);

comboBoxOperator.DataSource = employeeDataSet1.SelectAllUserNameFromOperator;

comboBoxOperator.DisplayMember = employeeDataSet1.SelectAllUserNameFromOperator.usernameColumn.ColumnName;

}

private void buttonLogin\_Click(object sender, EventArgs e)

{

selectOperatorTableAdapter1.Fill(employeeDataSet1.SelectOperator, comboBoxOperator.Text, textBoxPassword.Text);

if (employeeDataSet1.SelectOperator.Count == 0)

{

MessageBox.Show("操作员或密码不正确", "", MessageBoxButtons.OK, MessageBoxIcon.Error);

groupBox1.Enabled = false;

}

else

{

groupBox1.Enabled = true;

}

}

private void buttonExit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void radioButtonManage\_CheckedChanged(object sender, EventArgs e)

{

if (radioButtonManage.Checked == true)

{

this.Hide();

FormEmployee fm = new FormEmployee();

fm.ShowDialog();

this.Show();

}

}

private void radioButtonQuery\_CheckedChanged(object sender, EventArgs e)

{

if (radioButtonQuery.Checked == true)

{

int? maleNumber = 0;

int? femaleNumber = 0;

simpleQueryTableAdapter1.GetData(ref maleNumber , ref femaleNumber);

MessageBox.Show(string.Format("男：{0}，女：{1}", maleNumber, femaleNumber));

}

}

}

}

## [FromEmployee.cs]

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace BankEmployee

{

public partial class FormEmployee : Form

{

public FormEmployee()

{

InitializeComponent();

groupBox1.Text = "";

}

private void basicTableBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

try

{

this.Validate();

this.basicTableBindingSource.EndEdit();

this.basicTableTableAdapter.Update(this.employeeDataSet.BasicTable);

labelTip.Text = "保存成功";

}

catch(Exception err)

{

labelTip.Text = "保存失败：" + err.Message;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

this.basicTableTableAdapter.Fill(this.employeeDataSet.BasicTable);

ChangeCurrentPosition(CurrentOperator.当前记录);

}

private void buttonImport\_Click(object sender, EventArgs e)

{

OpenFileDialog openFileDialog1 = new OpenFileDialog();

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

Bitmap image = new Bitmap(openFileDialog1.FileName);

pictureBoxPhoto.Image = image;

}

}

private void buttonClear\_Click(object sender, EventArgs e)

{

pictureBoxPhoto.Image = null;

}

private void basicTableBindingNavigator\_RefreshItems(object sender, EventArgs e)

{

}

private void bindingNavigatorAddNewItem\_Click(object sender, EventArgs e)

{

try

{

basicTableBindingSource.AddNew();

ChangeCurrentPosition(CurrentOperator.当前记录);

textBoxName.Focus();

}

catch(Exception err)

{

labelTip.Text = "当前记录有错："+err.Message;

}

}

enum CurrentOperator { 第一个记录, 前一个记录, 下一个记录, 最后一个记录, 当前记录 };

private void ChangeCurrentPosition(CurrentOperator cp)

{

labelTip.Text = "温馨提示：操作一段时间后别忘了保存一下";

try

{

switch (cp)

{

case CurrentOperator.第一个记录:

basicTableBindingSource.MoveFirst();

break;

case CurrentOperator.前一个记录:

basicTableBindingSource.MovePrevious();

break;

case CurrentOperator.下一个记录:

basicTableBindingSource.MoveNext();

break;

case CurrentOperator.最后一个记录:

basicTableBindingSource.MoveLast();

break;

}

}

catch (Exception err)

{

labelTip.Text = "当前记录有错：" + err.Message;

}

int pos = basicTableBindingSource.Position;

int max=basicTableBindingSource.Count;

bindingNavigatorMoveFirstItem.Enabled = pos <= 0 ? false : true;

bindingNavigatorMovePreviousItem.Enabled = pos <= 0 ? false : true;

bindingNavigatorMoveNextItem.Enabled = pos >= max - 1 ? false : true;

bindingNavigatorMoveLastItem.Enabled = pos >= max - 1 ? false : true;

if (basicTableBindingSource.Count == 0)

{

labelTip.Text = "操作提示：库中无记录，请单击【+】添加新记录";

groupBox1.Enabled = false;

}

else

{

groupBox1.Enabled = true;

}

}

private void bindingNavigatorMovePreviousItem\_Click(object sender, EventArgs e)

{

ChangeCurrentPosition(CurrentOperator.前一个记录);

}

private void bindingNavigatorMoveNextItem\_Click(object sender, EventArgs e)

{

ChangeCurrentPosition(CurrentOperator.下一个记录);

}

private void bindingNavigatorMoveLastItem\_Click(object sender, EventArgs e)

{

ChangeCurrentPosition(CurrentOperator.最后一个记录);

}

private void bindingNavigatorMoveFirstItem\_Click(object sender, EventArgs e)

{

ChangeCurrentPosition(CurrentOperator.第一个记录);

}

private void bindingNavigatorDeleteItem\_Click(object sender, EventArgs e)

{

try

{

basicTableBindingSource.RemoveCurrent();

ChangeCurrentPosition(CurrentOperator.当前记录);

}

catch (Exception err)

{

labelTip.Text = "出错：" + err.Message;

}

}

}

}

## [EmployeeDateSet.cs]

namespace BankEmployee {

partial class EmployeeDataSet

{

partial class SimpleQueryDataTable

{

}

}

}

## [EmployeeDataSet.Designer.cs]

namespace BankEmployee {

using System;

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[Serializable()]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedDataSetSchema")]

[System.Xml.Serialization.XmlRootAttribute("EmployeeDataSet")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")]

public partial class EmployeeDataSet : System.Data.DataSet {

private BasicTableDataTable tableBasicTable;

private OperatorDataTable tableOperator;

private SelectAllUserNameFromOperatorDataTable tableSelectAllUserNameFromOperator;

private SelectOperatorDataTable tableSelectOperator;

private SimpleQueryDataTable tableSimpleQuery;

private System.Data.SchemaSerializationMode \_schemaSerializationMode = System.Data.SchemaSerializationMode.IncludeSchema;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public EmployeeDataSet() {

this.BeginInit();

this.InitClass();

System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler = new System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

base.Tables.CollectionChanged += schemaChangedHandler;

base.Relations.CollectionChanged += schemaChangedHandler;

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected EmployeeDataSet(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context, false) {

if ((this.IsBinarySerialized(info, context) == true)) {

this.InitVars(false);

System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler1 = new System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

this.Tables.CollectionChanged += schemaChangedHandler1;

this.Relations.CollectionChanged += schemaChangedHandler1;

return;

}

string strSchema = ((string)(info.GetValue("XmlSchema", typeof(string))));

if ((this.DetermineSchemaSerializationMode(info, context) == System.Data.SchemaSerializationMode.IncludeSchema)) {

System.Data.DataSet ds = new System.Data.DataSet();

ds.ReadXmlSchema(new System.Xml.XmlTextReader(new System.IO.StringReader(strSchema)));

if ((ds.Tables["BasicTable"] != null)) {

base.Tables.Add(new BasicTableDataTable(ds.Tables["BasicTable"]));

}

if ((ds.Tables["Operator"] != null)) {

base.Tables.Add(new OperatorDataTable(ds.Tables["Operator"]));

}

if ((ds.Tables["SelectAllUserNameFromOperator"] != null)) {

base.Tables.Add(new SelectAllUserNameFromOperatorDataTable(ds.Tables["SelectAllUserNameFromOperator"]));

}

if ((ds.Tables["SelectOperator"] != null)) {

base.Tables.Add(new SelectOperatorDataTable(ds.Tables["SelectOperator"]));

}

if ((ds.Tables["SimpleQuery"] != null)) {

base.Tables.Add(new SimpleQueryDataTable(ds.Tables["SimpleQuery"]));

}

this.DataSetName = ds.DataSetName;

this.Prefix = ds.Prefix;

this.Namespace = ds.Namespace;

this.Locale = ds.Locale;

this.CaseSensitive = ds.CaseSensitive;

this.EnforceConstraints = ds.EnforceConstraints;

this.Merge(ds, false, System.Data.MissingSchemaAction.Add);

this.InitVars();

}

else {

this.ReadXmlSchema(new System.Xml.XmlTextReader(new System.IO.StringReader(strSchema)));

}

this.GetSerializationData(info, context);

System.ComponentModel.CollectionChangeEventHandler schemaChangedHandler = new System.ComponentModel.CollectionChangeEventHandler(this.SchemaChanged);

base.Tables.CollectionChanged += schemaChangedHandler;

this.Relations.CollectionChanged += schemaChangedHandler;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

[System.ComponentModel.DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Content)]

public BasicTableDataTable BasicTable {

get {

return this.tableBasicTable;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

[System.ComponentModel.DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Content)]

public OperatorDataTable Operator {

get {

return this.tableOperator;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

[System.ComponentModel.DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Content)]

public SelectAllUserNameFromOperatorDataTable SelectAllUserNameFromOperator {

get {

return this.tableSelectAllUserNameFromOperator;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

[System.ComponentModel.DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Content)]

public SelectOperatorDataTable SelectOperator {

get {

return this.tableSelectOperator;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

[System.ComponentModel.DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Content)]

public SimpleQueryDataTable SimpleQuery {

get {

return this.tableSimpleQuery;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.BrowsableAttribute(true)]

[System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializationVisibility.Visible)]

public override System.Data.SchemaSerializationMode SchemaSerializationMode {

get {

return this.\_schemaSerializationMode;

}

set {

this.\_schemaSerializationMode = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializationVisibility.Hidden)]

public new System.Data.DataTableCollection Tables {

get {

return base.Tables;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.DesignerSerializationVisibilityAttribute(System.ComponentModel.DesignerSerializationVisibility.Hidden)]

public new System.Data.DataRelationCollection Relations {

get {

return base.Relations;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void InitializeDerivedDataSet() {

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataSet Clone() {

EmployeeDataSet cln = ((EmployeeDataSet)(base.Clone()));

cln.InitVars();

cln.SchemaSerializationMode = this.SchemaSerializationMode;

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override bool ShouldSerializeTables() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override bool ShouldSerializeRelations() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void ReadXmlSerializable(System.Xml.XmlReader reader) {

if ((this.DetermineSchemaSerializationMode(reader) == System.Data.SchemaSerializationMode.IncludeSchema)) {

this.Reset();

System.Data.DataSet ds = new System.Data.DataSet();

ds.ReadXml(reader);

if ((ds.Tables["BasicTable"] != null)) {

base.Tables.Add(new BasicTableDataTable(ds.Tables["BasicTable"]));

}

if ((ds.Tables["Operator"] != null)) {

base.Tables.Add(new OperatorDataTable(ds.Tables["Operator"]));

}

if ((ds.Tables["SelectAllUserNameFromOperator"] != null)) {

base.Tables.Add(new SelectAllUserNameFromOperatorDataTable(ds.Tables["SelectAllUserNameFromOperator"]));

}

if ((ds.Tables["SelectOperator"] != null)) {

base.Tables.Add(new SelectOperatorDataTable(ds.Tables["SelectOperator"]));

}

if ((ds.Tables["SimpleQuery"] != null)) {

base.Tables.Add(new SimpleQueryDataTable(ds.Tables["SimpleQuery"]));

}

this.DataSetName = ds.DataSetName;

this.Prefix = ds.Prefix;

this.Namespace = ds.Namespace;

this.Locale = ds.Locale;

this.CaseSensitive = ds.CaseSensitive;

this.EnforceConstraints = ds.EnforceConstraints;

this.Merge(ds, false, System.Data.MissingSchemaAction.Add);

this.InitVars();

}

else {

this.ReadXml(reader);

this.InitVars();

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Xml.Schema.XmlSchema GetSchemaSerializable() {

System.IO.MemoryStream stream = new System.IO.MemoryStream();

this.WriteXmlSchema(new System.Xml.XmlTextWriter(stream, null));

stream.Position = 0;

return System.Xml.Schema.XmlSchema.Read(new System.Xml.XmlTextReader(stream), null);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.InitVars(true);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars(bool initTable) {

this.tableBasicTable = ((BasicTableDataTable)(base.Tables["BasicTable"]));

if ((initTable == true)) {

if ((this.tableBasicTable != null)) {

this.tableBasicTable.InitVars();

}

}

this.tableOperator = ((OperatorDataTable)(base.Tables["Operator"]));

if ((initTable == true)) {

if ((this.tableOperator != null)) {

this.tableOperator.InitVars();

}

}

this.tableSelectAllUserNameFromOperator = ((SelectAllUserNameFromOperatorDataTable)(base.Tables["SelectAllUserNameFromOperator"]));

if ((initTable == true)) {

if ((this.tableSelectAllUserNameFromOperator != null)) {

this.tableSelectAllUserNameFromOperator.InitVars();

}

}

this.tableSelectOperator = ((SelectOperatorDataTable)(base.Tables["SelectOperator"]));

if ((initTable == true)) {

if ((this.tableSelectOperator != null)) {

this.tableSelectOperator.InitVars();

}

}

this.tableSimpleQuery = ((SimpleQueryDataTable)(base.Tables["SimpleQuery"]));

if ((initTable == true)) {

if ((this.tableSimpleQuery != null)) {

this.tableSimpleQuery.InitVars();

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.DataSetName = "EmployeeDataSet";

this.Prefix = "";

this.Namespace = "http://tempuri.org/EmployeeDataSet.xsd";

this.EnforceConstraints = true;

this.SchemaSerializationMode = System.Data.SchemaSerializationMode.IncludeSchema;

this.tableBasicTable = new BasicTableDataTable();

base.Tables.Add(this.tableBasicTable);

this.tableOperator = new OperatorDataTable();

base.Tables.Add(this.tableOperator);

this.tableSelectAllUserNameFromOperator = new SelectAllUserNameFromOperatorDataTable();

base.Tables.Add(this.tableSelectAllUserNameFromOperator);

this.tableSelectOperator = new SelectOperatorDataTable();

base.Tables.Add(this.tableSelectOperator);

this.tableSimpleQuery = new SimpleQueryDataTable();

base.Tables.Add(this.tableSimpleQuery);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private bool ShouldSerializeBasicTable() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private bool ShouldSerializeOperator() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private bool ShouldSerializeSelectAllUserNameFromOperator() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private bool ShouldSerializeSelectOperator() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private bool ShouldSerializeSimpleQuery() {

return false;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void SchemaChanged(object sender, System.ComponentModel.CollectionChangeEventArgs e) {

if ((e.Action == System.ComponentModel.CollectionChangeAction.Remove)) {

this.InitVars();

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedDataSetSchema(System.Xml.Schema.XmlSchemaSet xs) {

EmployeeDataSet ds = new EmployeeDataSet();

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any = new System.Xml.Schema.XmlSchemaAny();

any.Namespace = ds.Namespace;

sequence.Items.Add(any);

type.Particle = sequence;

return type;

}

public delegate void BasicTableRowChangeEventHandler(object sender, BasicTableRowChangeEvent e);

public delegate void OperatorRowChangeEventHandler(object sender, OperatorRowChangeEvent e);

public delegate void SelectAllUserNameFromOperatorRowChangeEventHandler(object sender, SelectAllUserNameFromOperatorRowChangeEvent e);

public delegate void SelectOperatorRowChangeEventHandler(object sender, SelectOperatorRowChangeEvent e);

public delegate void SimpleQueryRowChangeEventHandler(object sender, SimpleQueryRowChangeEvent e);

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.Serializable()]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]

public partial class BasicTableDataTable : System.Data.DataTable, System.Collections.IEnumerable {

private System.Data.DataColumn column职员号;

private System.Data.DataColumn column姓名;

private System.Data.DataColumn column性别;

private System.Data.DataColumn column出生日期;

private System.Data.DataColumn column地址;

private System.Data.DataColumn column照片;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableDataTable() {

this.TableName = "BasicTable";

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal BasicTableDataTable(System.Data.DataTable table) {

this.TableName = table.TableName;

if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {

this.CaseSensitive = table.CaseSensitive;

}

if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {

this.Locale = table.Locale;

}

if ((table.Namespace != table.DataSet.Namespace)) {

this.Namespace = table.Namespace;

}

this.Prefix = table.Prefix;

this.MinimumCapacity = table.MinimumCapacity;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected BasicTableDataTable(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context) {

this.InitVars();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 职员号Column {

get {

return this.column职员号;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 姓名Column {

get {

return this.column姓名;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 性别Column {

get {

return this.column性别;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 出生日期Column {

get {

return this.column出生日期;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 地址Column {

get {

return this.column地址;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn 照片Column {

get {

return this.column照片;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

public int Count {

get {

return this.Rows.Count;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRow this[int index] {

get {

return ((BasicTableRow)(this.Rows[index]));

}

}

public event BasicTableRowChangeEventHandler BasicTableRowChanging;

public event BasicTableRowChangeEventHandler BasicTableRowChanged;

public event BasicTableRowChangeEventHandler BasicTableRowDeleting;

public event BasicTableRowChangeEventHandler BasicTableRowDeleted;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void AddBasicTableRow(BasicTableRow row) {

this.Rows.Add(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRow AddBasicTableRow(string 职员号, string 姓名, string 性别, System.DateTime 出生日期, string 地址, byte[] 照片) {

BasicTableRow rowBasicTableRow = ((BasicTableRow)(this.NewRow()));

rowBasicTableRow.ItemArray = new object[] {

职员号,

姓名,

性别,

出生日期,

地址,

照片};

this.Rows.Add(rowBasicTableRow);

return rowBasicTableRow;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRow FindBy职员号(string 职员号) {

return ((BasicTableRow)(this.Rows.Find(new object[] {

职员号})));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public virtual System.Collections.IEnumerator GetEnumerator() {

return this.Rows.GetEnumerator();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataTable Clone() {

BasicTableDataTable cln = ((BasicTableDataTable)(base.Clone()));

cln.InitVars();

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataTable CreateInstance() {

return new BasicTableDataTable();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.column职员号 = base.Columns["职员号"];

this.column姓名 = base.Columns["姓名"];

this.column性别 = base.Columns["性别"];

this.column出生日期 = base.Columns["出生日期"];

this.column地址 = base.Columns["地址"];

this.column照片 = base.Columns["照片"];

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.column职员号 = new System.Data.DataColumn("职员号", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.column职员号);

this.column姓名 = new System.Data.DataColumn("姓名", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.column姓名);

this.column性别 = new System.Data.DataColumn("性别", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.column性别);

this.column出生日期 = new System.Data.DataColumn("出生日期", typeof(System.DateTime), null, System.Data.MappingType.Element);

base.Columns.Add(this.column出生日期);

this.column地址 = new System.Data.DataColumn("地址", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.column地址);

this.column照片 = new System.Data.DataColumn("照片", typeof(byte[]), null, System.Data.MappingType.Element);

base.Columns.Add(this.column照片);

this.Constraints.Add(new System.Data.UniqueConstraint("Constraint1", new System.Data.DataColumn[] {

this.column职员号}, true));

this.column职员号.AllowDBNull = false;

this.column职员号.Unique = true;

this.column职员号.MaxLength = 5;

this.column姓名.AllowDBNull = false;

this.column姓名.MaxLength = 10;

this.column性别.MaxLength = 1;

this.column地址.MaxLength = 50;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRow NewBasicTableRow() {

return ((BasicTableRow)(this.NewRow()));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataRow NewRowFromBuilder(System.Data.DataRowBuilder builder) {

return new BasicTableRow(builder);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Type GetRowType() {

return typeof(BasicTableRow);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanged(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanged(e);

if ((this.BasicTableRowChanged != null)) {

this.BasicTableRowChanged(this, new BasicTableRowChangeEvent(((BasicTableRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanging(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanging(e);

if ((this.BasicTableRowChanging != null)) {

this.BasicTableRowChanging(this, new BasicTableRowChangeEvent(((BasicTableRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleted(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleted(e);

if ((this.BasicTableRowDeleted != null)) {

this.BasicTableRowDeleted(this, new BasicTableRowChangeEvent(((BasicTableRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleting(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleting(e);

if ((this.BasicTableRowDeleting != null)) {

this.BasicTableRowDeleting(this, new BasicTableRowChangeEvent(((BasicTableRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void RemoveBasicTableRow(BasicTableRow row) {

this.Rows.Remove(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedTableSchema(System.Xml.Schema.XmlSchemaSet xs) {

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

EmployeeDataSet ds = new EmployeeDataSet();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any1 = new System.Xml.Schema.XmlSchemaAny();

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

System.Xml.Schema.XmlSchemaAny any2 = new System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

System.Xml.Schema.XmlSchemaAttribute attribute1 = new System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

System.Xml.Schema.XmlSchemaAttribute attribute2 = new System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableTypeName";

attribute2.FixedValue = "BasicTableDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

return type;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.Serializable()]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]

public partial class OperatorDataTable : System.Data.DataTable, System.Collections.IEnumerable {

private System.Data.DataColumn columnusername;

private System.Data.DataColumn columnpassword;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorDataTable() {

this.TableName = "Operator";

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal OperatorDataTable(System.Data.DataTable table) {

this.TableName = table.TableName;

if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {

this.CaseSensitive = table.CaseSensitive;

}

if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {

this.Locale = table.Locale;

}

if ((table.Namespace != table.DataSet.Namespace)) {

this.Namespace = table.Namespace;

}

this.Prefix = table.Prefix;

this.MinimumCapacity = table.MinimumCapacity;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected OperatorDataTable(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context) {

this.InitVars();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn usernameColumn {

get {

return this.columnusername;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn passwordColumn {

get {

return this.columnpassword;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

public int Count {

get {

return this.Rows.Count;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRow this[int index] {

get {

return ((OperatorRow)(this.Rows[index]));

}

}

public event OperatorRowChangeEventHandler OperatorRowChanging;

public event OperatorRowChangeEventHandler OperatorRowChanged;

public event OperatorRowChangeEventHandler OperatorRowDeleting;

public event OperatorRowChangeEventHandler OperatorRowDeleted;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void AddOperatorRow(OperatorRow row) {

this.Rows.Add(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRow AddOperatorRow(string username, string password) {

OperatorRow rowOperatorRow = ((OperatorRow)(this.NewRow()));

rowOperatorRow.ItemArray = new object[] {

username,

password};

this.Rows.Add(rowOperatorRow);

return rowOperatorRow;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRow FindByusername(string username) {

return ((OperatorRow)(this.Rows.Find(new object[] {

username})));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public virtual System.Collections.IEnumerator GetEnumerator() {

return this.Rows.GetEnumerator();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataTable Clone() {

OperatorDataTable cln = ((OperatorDataTable)(base.Clone()));

cln.InitVars();

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataTable CreateInstance() {

return new OperatorDataTable();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.columnusername = base.Columns["username"];

this.columnpassword = base.Columns["password"];

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.columnusername = new System.Data.DataColumn("username", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.columnusername);

this.columnpassword = new System.Data.DataColumn("password", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.columnpassword);

this.Constraints.Add(new System.Data.UniqueConstraint("Constraint1", new System.Data.DataColumn[] {

this.columnusername}, true));

this.columnusername.AllowDBNull = false;

this.columnusername.Unique = true;

this.columnusername.MaxLength = 10;

this.columnpassword.MaxLength = 20;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRow NewOperatorRow() {

return ((OperatorRow)(this.NewRow()));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataRow NewRowFromBuilder(System.Data.DataRowBuilder builder) {

return new OperatorRow(builder);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Type GetRowType() {

return typeof(OperatorRow);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanged(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanged(e);

if ((this.OperatorRowChanged != null)) {

this.OperatorRowChanged(this, new OperatorRowChangeEvent(((OperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanging(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanging(e);

if ((this.OperatorRowChanging != null)) {

this.OperatorRowChanging(this, new OperatorRowChangeEvent(((OperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleted(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleted(e);

if ((this.OperatorRowDeleted != null)) {

this.OperatorRowDeleted(this, new OperatorRowChangeEvent(((OperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleting(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleting(e);

if ((this.OperatorRowDeleting != null)) {

this.OperatorRowDeleting(this, new OperatorRowChangeEvent(((OperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void RemoveOperatorRow(OperatorRow row) {

this.Rows.Remove(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedTableSchema(System.Xml.Schema.XmlSchemaSet xs) {

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

EmployeeDataSet ds = new EmployeeDataSet();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any1 = new System.Xml.Schema.XmlSchemaAny();

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

System.Xml.Schema.XmlSchemaAny any2 = new System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

System.Xml.Schema.XmlSchemaAttribute attribute1 = new System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

System.Xml.Schema.XmlSchemaAttribute attribute2 = new System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableTypeName";

attribute2.FixedValue = "OperatorDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

return type;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.Serializable()]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]

public partial class SelectAllUserNameFromOperatorDataTable : System.Data.DataTable, System.Collections.IEnumerable {

private System.Data.DataColumn columnusername;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorDataTable() {

this.TableName = "SelectAllUserNameFromOperator";

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SelectAllUserNameFromOperatorDataTable(System.Data.DataTable table) {

this.TableName = table.TableName;

if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {

this.CaseSensitive = table.CaseSensitive;

}

if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {

this.Locale = table.Locale;

}

if ((table.Namespace != table.DataSet.Namespace)) {

this.Namespace = table.Namespace;

}

this.Prefix = table.Prefix;

this.MinimumCapacity = table.MinimumCapacity;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected SelectAllUserNameFromOperatorDataTable(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context) {

this.InitVars();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn usernameColumn {

get {

return this.columnusername;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

public int Count {

get {

return this.Rows.Count;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRow this[int index] {

get {

return ((SelectAllUserNameFromOperatorRow)(this.Rows[index]));

}

}

public event SelectAllUserNameFromOperatorRowChangeEventHandler SelectAllUserNameFromOperatorRowChanging;

public event SelectAllUserNameFromOperatorRowChangeEventHandler SelectAllUserNameFromOperatorRowChanged;

public event SelectAllUserNameFromOperatorRowChangeEventHandler SelectAllUserNameFromOperatorRowDeleting;

public event SelectAllUserNameFromOperatorRowChangeEventHandler SelectAllUserNameFromOperatorRowDeleted;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void AddSelectAllUserNameFromOperatorRow(SelectAllUserNameFromOperatorRow row) {

this.Rows.Add(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRow AddSelectAllUserNameFromOperatorRow(string username) {

SelectAllUserNameFromOperatorRow rowSelectAllUserNameFromOperatorRow = ((SelectAllUserNameFromOperatorRow)(this.NewRow()));

rowSelectAllUserNameFromOperatorRow.ItemArray = new object[] {

username};

this.Rows.Add(rowSelectAllUserNameFromOperatorRow);

return rowSelectAllUserNameFromOperatorRow;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRow FindByusername(string username) {

return ((SelectAllUserNameFromOperatorRow)(this.Rows.Find(new object[] {

username})));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public virtual System.Collections.IEnumerator GetEnumerator() {

return this.Rows.GetEnumerator();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataTable Clone() {

SelectAllUserNameFromOperatorDataTable cln = ((SelectAllUserNameFromOperatorDataTable)(base.Clone()));

cln.InitVars();

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataTable CreateInstance() {

return new SelectAllUserNameFromOperatorDataTable();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.columnusername = base.Columns["username"];

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.columnusername = new System.Data.DataColumn("username", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.columnusername);

this.Constraints.Add(new System.Data.UniqueConstraint("Constraint1", new System.Data.DataColumn[] {

this.columnusername}, true));

this.columnusername.AllowDBNull = false;

this.columnusername.Unique = true;

this.columnusername.MaxLength = 10;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRow NewSelectAllUserNameFromOperatorRow() {

return ((SelectAllUserNameFromOperatorRow)(this.NewRow()));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataRow NewRowFromBuilder(System.Data.DataRowBuilder builder) {

return new SelectAllUserNameFromOperatorRow(builder);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Type GetRowType() {

return typeof(SelectAllUserNameFromOperatorRow);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanged(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanged(e);

if ((this.SelectAllUserNameFromOperatorRowChanged != null)) {

this.SelectAllUserNameFromOperatorRowChanged(this, new SelectAllUserNameFromOperatorRowChangeEvent(((SelectAllUserNameFromOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanging(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanging(e);

if ((this.SelectAllUserNameFromOperatorRowChanging != null)) {

this.SelectAllUserNameFromOperatorRowChanging(this, new SelectAllUserNameFromOperatorRowChangeEvent(((SelectAllUserNameFromOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleted(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleted(e);

if ((this.SelectAllUserNameFromOperatorRowDeleted != null)) {

this.SelectAllUserNameFromOperatorRowDeleted(this, new SelectAllUserNameFromOperatorRowChangeEvent(((SelectAllUserNameFromOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleting(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleting(e);

if ((this.SelectAllUserNameFromOperatorRowDeleting != null)) {

this.SelectAllUserNameFromOperatorRowDeleting(this, new SelectAllUserNameFromOperatorRowChangeEvent(((SelectAllUserNameFromOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void RemoveSelectAllUserNameFromOperatorRow(SelectAllUserNameFromOperatorRow row) {

this.Rows.Remove(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedTableSchema(System.Xml.Schema.XmlSchemaSet xs) {

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

EmployeeDataSet ds = new EmployeeDataSet();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any1 = new System.Xml.Schema.XmlSchemaAny();

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

System.Xml.Schema.XmlSchemaAny any2 = new System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

System.Xml.Schema.XmlSchemaAttribute attribute1 = new System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

System.Xml.Schema.XmlSchemaAttribute attribute2 = new System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableTypeName";

attribute2.FixedValue = "SelectAllUserNameFromOperatorDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

return type;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.Serializable()]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]

public partial class SelectOperatorDataTable : System.Data.DataTable, System.Collections.IEnumerable {

private System.Data.DataColumn columnusername;

private System.Data.DataColumn columnpassword;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorDataTable() {

this.TableName = "SelectOperator";

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SelectOperatorDataTable(System.Data.DataTable table) {

this.TableName = table.TableName;

if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {

this.CaseSensitive = table.CaseSensitive;

}

if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {

this.Locale = table.Locale;

}

if ((table.Namespace != table.DataSet.Namespace)) {

this.Namespace = table.Namespace;

}

this.Prefix = table.Prefix;

this.MinimumCapacity = table.MinimumCapacity;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected SelectOperatorDataTable(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context) {

this.InitVars();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn usernameColumn {

get {

return this.columnusername;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn passwordColumn {

get {

return this.columnpassword;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

public int Count {

get {

return this.Rows.Count;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRow this[int index] {

get {

return ((SelectOperatorRow)(this.Rows[index]));

}

}

public event SelectOperatorRowChangeEventHandler SelectOperatorRowChanging;

public event SelectOperatorRowChangeEventHandler SelectOperatorRowChanged;

public event SelectOperatorRowChangeEventHandler SelectOperatorRowDeleting;

public event SelectOperatorRowChangeEventHandler SelectOperatorRowDeleted;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void AddSelectOperatorRow(SelectOperatorRow row) {

this.Rows.Add(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRow AddSelectOperatorRow(string username, string password) {

SelectOperatorRow rowSelectOperatorRow = ((SelectOperatorRow)(this.NewRow()));

rowSelectOperatorRow.ItemArray = new object[] {

username,

password};

this.Rows.Add(rowSelectOperatorRow);

return rowSelectOperatorRow;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRow FindByusername(string username) {

return ((SelectOperatorRow)(this.Rows.Find(new object[] {

username})));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public virtual System.Collections.IEnumerator GetEnumerator() {

return this.Rows.GetEnumerator();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataTable Clone() {

SelectOperatorDataTable cln = ((SelectOperatorDataTable)(base.Clone()));

cln.InitVars();

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataTable CreateInstance() {

return new SelectOperatorDataTable();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.columnusername = base.Columns["username"];

this.columnpassword = base.Columns["password"];

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.columnusername = new System.Data.DataColumn("username", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.columnusername);

this.columnpassword = new System.Data.DataColumn("password", typeof(string), null, System.Data.MappingType.Element);

base.Columns.Add(this.columnpassword);

this.Constraints.Add(new System.Data.UniqueConstraint("Constraint1", new System.Data.DataColumn[] {

this.columnusername}, true));

this.columnusername.AllowDBNull = false;

this.columnusername.Unique = true;

this.columnusername.MaxLength = 10;

this.columnpassword.MaxLength = 20;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRow NewSelectOperatorRow() {

return ((SelectOperatorRow)(this.NewRow()));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataRow NewRowFromBuilder(System.Data.DataRowBuilder builder) {

return new SelectOperatorRow(builder);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Type GetRowType() {

return typeof(SelectOperatorRow);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanged(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanged(e);

if ((this.SelectOperatorRowChanged != null)) {

this.SelectOperatorRowChanged(this, new SelectOperatorRowChangeEvent(((SelectOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanging(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanging(e);

if ((this.SelectOperatorRowChanging != null)) {

this.SelectOperatorRowChanging(this, new SelectOperatorRowChangeEvent(((SelectOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleted(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleted(e);

if ((this.SelectOperatorRowDeleted != null)) {

this.SelectOperatorRowDeleted(this, new SelectOperatorRowChangeEvent(((SelectOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleting(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleting(e);

if ((this.SelectOperatorRowDeleting != null)) {

this.SelectOperatorRowDeleting(this, new SelectOperatorRowChangeEvent(((SelectOperatorRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void RemoveSelectOperatorRow(SelectOperatorRow row) {

this.Rows.Remove(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedTableSchema(System.Xml.Schema.XmlSchemaSet xs) {

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

EmployeeDataSet ds = new EmployeeDataSet();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any1 = new System.Xml.Schema.XmlSchemaAny();

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

System.Xml.Schema.XmlSchemaAny any2 = new System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

System.Xml.Schema.XmlSchemaAttribute attribute1 = new System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

System.Xml.Schema.XmlSchemaAttribute attribute2 = new System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableTypeName";

attribute2.FixedValue = "SelectOperatorDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

return type;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.Serializable()]

[System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedTableSchema")]

public partial class SimpleQueryDataTable : System.Data.DataTable, System.Collections.IEnumerable {

private System.Data.DataColumn \_column\_maleNumber;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryDataTable() {

this.TableName = "SimpleQuery";

this.BeginInit();

this.InitClass();

this.EndInit();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SimpleQueryDataTable(System.Data.DataTable table) {

this.TableName = table.TableName;

if ((table.CaseSensitive != table.DataSet.CaseSensitive)) {

this.CaseSensitive = table.CaseSensitive;

}

if ((table.Locale.ToString() != table.DataSet.Locale.ToString())) {

this.Locale = table.Locale;

}

if ((table.Namespace != table.DataSet.Namespace)) {

this.Namespace = table.Namespace;

}

this.Prefix = table.Prefix;

this.MinimumCapacity = table.MinimumCapacity;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected SimpleQueryDataTable(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context) :

base(info, context) {

this.InitVars();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataColumn @maleNumberColumn {

get {

return this.\_column\_maleNumber;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Browsable(false)]

public int Count {

get {

return this.Rows.Count;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryRow this[int index] {

get {

return ((SimpleQueryRow)(this.Rows[index]));

}

}

public event SimpleQueryRowChangeEventHandler SimpleQueryRowChanging;

public event SimpleQueryRowChangeEventHandler SimpleQueryRowChanged;

public event SimpleQueryRowChangeEventHandler SimpleQueryRowDeleting;

public event SimpleQueryRowChangeEventHandler SimpleQueryRowDeleted;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void AddSimpleQueryRow(SimpleQueryRow row) {

this.Rows.Add(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryRow AddSimpleQueryRow(int @maleNumber) {

SimpleQueryRow rowSimpleQueryRow = ((SimpleQueryRow)(this.NewRow()));

rowSimpleQueryRow.ItemArray = new object[] {

@maleNumber};

this.Rows.Add(rowSimpleQueryRow);

return rowSimpleQueryRow;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public virtual System.Collections.IEnumerator GetEnumerator() {

return this.Rows.GetEnumerator();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public override System.Data.DataTable Clone() {

SimpleQueryDataTable cln = ((SimpleQueryDataTable)(base.Clone()));

cln.InitVars();

return cln;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataTable CreateInstance() {

return new SimpleQueryDataTable();

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal void InitVars() {

this.\_column\_maleNumber = base.Columns["@maleNumber"];

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitClass() {

this.\_column\_maleNumber = new System.Data.DataColumn("@maleNumber", typeof(int), null, System.Data.MappingType.Element);

this.\_column\_maleNumber.ExtendedProperties.Add("Generator\_ColumnVarNameInTable", "\_column\_maleNumber");

this.\_column\_maleNumber.ExtendedProperties.Add("Generator\_UserColumnName", "@maleNumber");

base.Columns.Add(this.\_column\_maleNumber);

this.\_column\_maleNumber.ReadOnly = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryRow NewSimpleQueryRow() {

return ((SimpleQueryRow)(this.NewRow()));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Data.DataRow NewRowFromBuilder(System.Data.DataRowBuilder builder) {

return new SimpleQueryRow(builder);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override System.Type GetRowType() {

return typeof(SimpleQueryRow);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanged(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanged(e);

if ((this.SimpleQueryRowChanged != null)) {

this.SimpleQueryRowChanged(this, new SimpleQueryRowChangeEvent(((SimpleQueryRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowChanging(System.Data.DataRowChangeEventArgs e) {

base.OnRowChanging(e);

if ((this.SimpleQueryRowChanging != null)) {

this.SimpleQueryRowChanging(this, new SimpleQueryRowChangeEvent(((SimpleQueryRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleted(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleted(e);

if ((this.SimpleQueryRowDeleted != null)) {

this.SimpleQueryRowDeleted(this, new SimpleQueryRowChangeEvent(((SimpleQueryRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected override void OnRowDeleting(System.Data.DataRowChangeEventArgs e) {

base.OnRowDeleting(e);

if ((this.SimpleQueryRowDeleting != null)) {

this.SimpleQueryRowDeleting(this, new SimpleQueryRowChangeEvent(((SimpleQueryRow)(e.Row)), e.Action));

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void RemoveSimpleQueryRow(SimpleQueryRow row) {

this.Rows.Remove(row);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public static System.Xml.Schema.XmlSchemaComplexType GetTypedTableSchema(System.Xml.Schema.XmlSchemaSet xs) {

System.Xml.Schema.XmlSchemaComplexType type = new System.Xml.Schema.XmlSchemaComplexType();

System.Xml.Schema.XmlSchemaSequence sequence = new System.Xml.Schema.XmlSchemaSequence();

EmployeeDataSet ds = new EmployeeDataSet();

xs.Add(ds.GetSchemaSerializable());

System.Xml.Schema.XmlSchemaAny any1 = new System.Xml.Schema.XmlSchemaAny();

any1.Namespace = "http://www.w3.org/2001/XMLSchema";

any1.MinOccurs = new decimal(0);

any1.MaxOccurs = decimal.MaxValue;

any1.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any1);

System.Xml.Schema.XmlSchemaAny any2 = new System.Xml.Schema.XmlSchemaAny();

any2.Namespace = "urn:schemas-microsoft-com:xml-diffgram-v1";

any2.MinOccurs = new decimal(1);

any2.ProcessContents = System.Xml.Schema.XmlSchemaContentProcessing.Lax;

sequence.Items.Add(any2);

System.Xml.Schema.XmlSchemaAttribute attribute1 = new System.Xml.Schema.XmlSchemaAttribute();

attribute1.Name = "namespace";

attribute1.FixedValue = ds.Namespace;

type.Attributes.Add(attribute1);

System.Xml.Schema.XmlSchemaAttribute attribute2 = new System.Xml.Schema.XmlSchemaAttribute();

attribute2.Name = "tableTypeName";

attribute2.FixedValue = "SimpleQueryDataTable";

type.Attributes.Add(attribute2);

type.Particle = sequence;

return type;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public partial class BasicTableRow : System.Data.DataRow {

private BasicTableDataTable tableBasicTable;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal BasicTableRow(System.Data.DataRowBuilder rb) :

base(rb) {

this.tableBasicTable = ((BasicTableDataTable)(this.Table));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string 职员号 {

get {

return ((string)(this[this.tableBasicTable.职员号Column]));

}

set {

this[this.tableBasicTable.职员号Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string 姓名 {

get {

return ((string)(this[this.tableBasicTable.姓名Column]));

}

set {

this[this.tableBasicTable.姓名Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string 性别 {

get {

try {

return ((string)(this[this.tableBasicTable.性别Column]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“BasicTable”中列“性别”的值为 DBNull。", e);

}

}

set {

this[this.tableBasicTable.性别Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.DateTime 出生日期 {

get {

try {

return ((System.DateTime)(this[this.tableBasicTable.出生日期Column]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“BasicTable”中列“出生日期”的值为 DBNull。", e);

}

}

set {

this[this.tableBasicTable.出生日期Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string 地址 {

get {

try {

return ((string)(this[this.tableBasicTable.地址Column]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“BasicTable”中列“地址”的值为 DBNull。", e);

}

}

set {

this[this.tableBasicTable.地址Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public byte[] 照片 {

get {

try {

return ((byte[])(this[this.tableBasicTable.照片Column]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“BasicTable”中列“照片”的值为 DBNull。", e);

}

}

set {

this[this.tableBasicTable.照片Column] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool Is性别Null() {

return this.IsNull(this.tableBasicTable.性别Column);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void Set性别Null() {

this[this.tableBasicTable.性别Column] = System.Convert.DBNull;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool Is出生日期Null() {

return this.IsNull(this.tableBasicTable.出生日期Column);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void Set出生日期Null() {

this[this.tableBasicTable.出生日期Column] = System.Convert.DBNull;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool Is地址Null() {

return this.IsNull(this.tableBasicTable.地址Column);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void Set地址Null() {

this[this.tableBasicTable.地址Column] = System.Convert.DBNull;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool Is照片Null() {

return this.IsNull(this.tableBasicTable.照片Column);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void Set照片Null() {

this[this.tableBasicTable.照片Column] = System.Convert.DBNull;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public partial class OperatorRow : System.Data.DataRow {

private OperatorDataTable tableOperator;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal OperatorRow(System.Data.DataRowBuilder rb) :

base(rb) {

this.tableOperator = ((OperatorDataTable)(this.Table));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string username {

get {

return ((string)(this[this.tableOperator.usernameColumn]));

}

set {

this[this.tableOperator.usernameColumn] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string password {

get {

try {

return ((string)(this[this.tableOperator.passwordColumn]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“Operator”中列“password”的值为 DBNull。", e);

}

}

set {

this[this.tableOperator.passwordColumn] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool IspasswordNull() {

return this.IsNull(this.tableOperator.passwordColumn);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void SetpasswordNull() {

this[this.tableOperator.passwordColumn] = System.Convert.DBNull;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public partial class SelectAllUserNameFromOperatorRow : System.Data.DataRow {

private SelectAllUserNameFromOperatorDataTable tableSelectAllUserNameFromOperator;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SelectAllUserNameFromOperatorRow(System.Data.DataRowBuilder rb) :

base(rb) {

this.tableSelectAllUserNameFromOperator = ((SelectAllUserNameFromOperatorDataTable)(this.Table));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string username {

get {

return ((string)(this[this.tableSelectAllUserNameFromOperator.usernameColumn]));

}

set {

this[this.tableSelectAllUserNameFromOperator.usernameColumn] = value;

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public partial class SelectOperatorRow : System.Data.DataRow {

private SelectOperatorDataTable tableSelectOperator;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SelectOperatorRow(System.Data.DataRowBuilder rb) :

base(rb) {

this.tableSelectOperator = ((SelectOperatorDataTable)(this.Table));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string username {

get {

return ((string)(this[this.tableSelectOperator.usernameColumn]));

}

set {

this[this.tableSelectOperator.usernameColumn] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public string password {

get {

try {

return ((string)(this[this.tableSelectOperator.passwordColumn]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“SelectOperator”中列“password”的值为 DBNull。", e);

}

}

set {

this[this.tableSelectOperator.passwordColumn] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool IspasswordNull() {

return this.IsNull(this.tableSelectOperator.passwordColumn);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void SetpasswordNull() {

this[this.tableSelectOperator.passwordColumn] = System.Convert.DBNull;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public partial class SimpleQueryRow : System.Data.DataRow {

private SimpleQueryDataTable tableSimpleQuery;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal SimpleQueryRow(System.Data.DataRowBuilder rb) :

base(rb) {

this.tableSimpleQuery = ((SimpleQueryDataTable)(this.Table));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public int @maleNumber {

get {

try {

return ((int)(this[this.tableSimpleQuery.@maleNumberColumn]));

}

catch (System.InvalidCastException e) {

throw new System.Data.StrongTypingException("表“SimpleQuery”中列“@maleNumber”的值为 DBNull。", e);

}

}

set {

this[this.tableSimpleQuery.@maleNumberColumn] = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool \_Is\_maleNumberNull() {

return this.IsNull(this.tableSimpleQuery.@maleNumberColumn);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public void \_Set\_maleNumberNull() {

this[this.tableSimpleQuery.@maleNumberColumn] = System.Convert.DBNull;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public class BasicTableRowChangeEvent : System.EventArgs {

private BasicTableRow eventRow;

private System.Data.DataRowAction eventAction;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRowChangeEvent(BasicTableRow row, System.Data.DataRowAction action) {

this.eventRow = row;

this.eventAction = action;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableRow Row {

get {

return this.eventRow;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataRowAction Action {

get {

return this.eventAction;

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public class OperatorRowChangeEvent : System.EventArgs {

private OperatorRow eventRow;

private System.Data.DataRowAction eventAction;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRowChangeEvent(OperatorRow row, System.Data.DataRowAction action) {

this.eventRow = row;

this.eventAction = action;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorRow Row {

get {

return this.eventRow;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataRowAction Action {

get {

return this.eventAction;

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public class SelectAllUserNameFromOperatorRowChangeEvent : System.EventArgs {

private SelectAllUserNameFromOperatorRow eventRow;

private System.Data.DataRowAction eventAction;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRowChangeEvent(SelectAllUserNameFromOperatorRow row, System.Data.DataRowAction action) {

this.eventRow = row;

this.eventAction = action;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorRow Row {

get {

return this.eventRow;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataRowAction Action {

get {

return this.eventAction;

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public class SelectOperatorRowChangeEvent : System.EventArgs {

private SelectOperatorRow eventRow;

private System.Data.DataRowAction eventAction;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRowChangeEvent(SelectOperatorRow row, System.Data.DataRowAction action) {

this.eventRow = row;

this.eventAction = action;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorRow Row {

get {

return this.eventRow;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataRowAction Action {

get {

return this.eventAction;

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

public class SimpleQueryRowChangeEvent : System.EventArgs {

private SimpleQueryRow eventRow;

private System.Data.DataRowAction eventAction;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryRowChangeEvent(SimpleQueryRow row, System.Data.DataRowAction action) {

this.eventRow = row;

this.eventAction = action;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryRow Row {

get {

return this.eventRow;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public System.Data.DataRowAction Action {

get {

return this.eventAction;

}

}

}

}

}

namespace BankEmployee.EmployeeDataSetTableAdapters {

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.ComponentModel.DataObjectAttribute(true)]

[System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +

", Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class BasicTableTableAdapter : System.ComponentModel.Component {

private System.Data.SqlClient.SqlDataAdapter \_adapter;

private System.Data.SqlClient.SqlConnection \_connection;

private System.Data.SqlClient.SqlCommand[] \_commandCollection;

private bool \_clearBeforeFill;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public BasicTableTableAdapter() {

this.ClearBeforeFill = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private System.Data.SqlClient.SqlDataAdapter Adapter {

get {

if ((this.\_adapter == null)) {

this.InitAdapter();

}

return this.\_adapter;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal System.Data.SqlClient.SqlConnection Connection {

get {

if ((this.\_connection == null)) {

this.InitConnection();

}

return this.\_connection;

}

set {

this.\_connection = value;

if ((this.Adapter.InsertCommand != null)) {

this.Adapter.InsertCommand.Connection = value;

}

if ((this.Adapter.DeleteCommand != null)) {

this.Adapter.DeleteCommand.Connection = value;

}

if ((this.Adapter.UpdateCommand != null)) {

this.Adapter.UpdateCommand.Connection = value;

}

for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {

if ((this.CommandCollection[i] != null)) {

((System.Data.SqlClient.SqlCommand)(this.CommandCollection[i])).Connection = value;

}

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected System.Data.SqlClient.SqlCommand[] CommandCollection {

get {

if ((this.\_commandCollection == null)) {

this.InitCommandCollection();

}

return this.\_commandCollection;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool ClearBeforeFill {

get {

return this.\_clearBeforeFill;

}

set {

this.\_clearBeforeFill = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitAdapter() {

this.\_adapter = new System.Data.SqlClient.SqlDataAdapter();

System.Data.Common.DataTableMapping tableMapping = new System.Data.Common.DataTableMapping();

tableMapping.SourceTable = "Table";

tableMapping.DataSetTable = "BasicTable";

tableMapping.ColumnMappings.Add("职员号", "职员号");

tableMapping.ColumnMappings.Add("姓名", "姓名");

tableMapping.ColumnMappings.Add("性别", "性别");

tableMapping.ColumnMappings.Add("出生日期", "出生日期");

tableMapping.ColumnMappings.Add("地址", "地址");

tableMapping.ColumnMappings.Add("照片", "照片");

this.\_adapter.TableMappings.Add(tableMapping);

this.\_adapter.DeleteCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.DeleteCommand.Connection = this.Connection;

this.\_adapter.DeleteCommand.CommandText = @"DELETE FROM [dbo].[BasicTable] WHERE (([职员号] = @Original\_职员号) AND ([姓名] = @Original\_姓名) AND ((@IsNull\_性别 = 1 AND [性别] IS NULL) OR ([性别] = @Original\_性别)) AND ((@IsNull\_出生日期 = 1 AND [出生日期] IS NULL) OR ([出生日期] = @Original\_出生日期)) AND ((@IsNull\_地址 = 1 AND [地址] IS NULL) OR ([地址] = @Original\_地址)))";

this.\_adapter.DeleteCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_职员号", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "职员号", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_姓名", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "姓名", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_性别", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_性别", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_出生日期", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_出生日期", System.Data.SqlDbType.DateTime, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_地址", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_地址", System.Data.SqlDbType.NVarChar, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.InsertCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.InsertCommand.Connection = this.Connection;

this.\_adapter.InsertCommand.CommandText = "INSERT INTO [dbo].[BasicTable] ([职员号], [姓名], [性别], [出生日期], [地址], [照片]) VALUES (@职" +

"员号, @姓名, @性别, @出生日期, @地址, @照片);\r\nSELECT 职员号, 姓名, 性别, 出生日期, 地址, 照片 FROM BasicTabl" +

"e WHERE (职员号 = @职员号)";

this.\_adapter.InsertCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@职员号", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "职员号", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@姓名", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "姓名", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@性别", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@出生日期", System.Data.SqlDbType.DateTime, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@地址", System.Data.SqlDbType.NVarChar, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@照片", System.Data.SqlDbType.Image, 0, System.Data.ParameterDirection.Input, 0, 0, "照片", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.UpdateCommand.Connection = this.Connection;

this.\_adapter.UpdateCommand.CommandText = @"UPDATE [dbo].[BasicTable] SET [职员号] = @职员号, [姓名] = @姓名, [性别] = @性别, [出生日期] = @出生日期, [地址] = @地址, [照片] = @照片 WHERE (([职员号] = @Original\_职员号) AND ([姓名] = @Original\_姓名) AND ((@IsNull\_性别 = 1 AND [性别] IS NULL) OR ([性别] = @Original\_性别)) AND ((@IsNull\_出生日期 = 1 AND [出生日期] IS NULL) OR ([出生日期] = @Original\_出生日期)) AND ((@IsNull\_地址 = 1 AND [地址] IS NULL) OR ([地址] = @Original\_地址)));

SELECT 职员号, 姓名, 性别, 出生日期, 地址, 照片 FROM BasicTable WHERE (职员号 = @职员号)";

this.\_adapter.UpdateCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@职员号", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "职员号", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@姓名", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "姓名", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@性别", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@出生日期", System.Data.SqlDbType.DateTime, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@地址", System.Data.SqlDbType.NVarChar, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@照片", System.Data.SqlDbType.Image, 0, System.Data.ParameterDirection.Input, 0, 0, "照片", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_职员号", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "职员号", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_姓名", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "姓名", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_性别", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_性别", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "性别", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_出生日期", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_出生日期", System.Data.SqlDbType.DateTime, 0, System.Data.ParameterDirection.Input, 0, 0, "出生日期", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_地址", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_地址", System.Data.SqlDbType.NVarChar, 0, System.Data.ParameterDirection.Input, 0, 0, "地址", System.Data.DataRowVersion.Original, false, null, "", "", ""));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitConnection() {

this.\_connection = new System.Data.SqlClient.SqlConnection();

this.\_connection.ConnectionString = global::BankEmployee.Properties.Settings.Default.EmployeeConnectionString;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitCommandCollection() {

this.\_commandCollection = new System.Data.SqlClient.SqlCommand[1];

this.\_commandCollection[0] = new System.Data.SqlClient.SqlCommand();

this.\_commandCollection[0].Connection = this.Connection;

this.\_commandCollection[0].CommandText = "SELECT 职员号, 姓名, 性别, 出生日期, 地址, 照片 FROM dbo.BasicTable";

this.\_commandCollection[0].CommandType = System.Data.CommandType.Text;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Fill, true)]

public virtual int Fill(EmployeeDataSet.BasicTableDataTable dataTable) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((this.ClearBeforeFill == true)) {

dataTable.Clear();

}

int returnValue = this.Adapter.Fill(dataTable);

return returnValue;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, true)]

public virtual EmployeeDataSet.BasicTableDataTable GetData() {

this.Adapter.SelectCommand = this.CommandCollection[0];

EmployeeDataSet.BasicTableDataTable dataTable = new EmployeeDataSet.BasicTableDataTable();

this.Adapter.Fill(dataTable);

return dataTable;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(EmployeeDataSet.BasicTableDataTable dataTable) {

return this.Adapter.Update(dataTable);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(EmployeeDataSet dataSet) {

return this.Adapter.Update(dataSet, "BasicTable");

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(System.Data.DataRow dataRow) {

return this.Adapter.Update(new System.Data.DataRow[] {

dataRow});

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(System.Data.DataRow[] dataRows) {

return this.Adapter.Update(dataRows);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Delete, true)]

public virtual int Delete(string Original\_职员号, string Original\_姓名, string Original\_性别, System.Nullable<System.DateTime> Original\_出生日期, string Original\_地址) {

if ((Original\_职员号 == null)) {

throw new System.ArgumentNullException("Original\_职员号");

}

else {

this.Adapter.DeleteCommand.Parameters[0].Value = ((string)(Original\_职员号));

}

if ((Original\_姓名 == null)) {

throw new System.ArgumentNullException("Original\_姓名");

}

else {

this.Adapter.DeleteCommand.Parameters[1].Value = ((string)(Original\_姓名));

}

if ((Original\_性别 == null)) {

this.Adapter.DeleteCommand.Parameters[2].Value = ((object)(1));

this.Adapter.DeleteCommand.Parameters[3].Value = System.DBNull.Value;

}

else {

this.Adapter.DeleteCommand.Parameters[2].Value = ((object)(0));

this.Adapter.DeleteCommand.Parameters[3].Value = ((string)(Original\_性别));

}

if ((Original\_出生日期.HasValue == true)) {

this.Adapter.DeleteCommand.Parameters[4].Value = ((object)(0));

this.Adapter.DeleteCommand.Parameters[5].Value = ((System.DateTime)(Original\_出生日期.Value));

}

else {

this.Adapter.DeleteCommand.Parameters[4].Value = ((object)(1));

this.Adapter.DeleteCommand.Parameters[5].Value = System.DBNull.Value;

}

if ((Original\_地址 == null)) {

this.Adapter.DeleteCommand.Parameters[6].Value = ((object)(1));

this.Adapter.DeleteCommand.Parameters[7].Value = System.DBNull.Value;

}

else {

this.Adapter.DeleteCommand.Parameters[6].Value = ((object)(0));

this.Adapter.DeleteCommand.Parameters[7].Value = ((string)(Original\_地址));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.DeleteCommand.Connection.State;

if (((this.Adapter.DeleteCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.DeleteCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.DeleteCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.DeleteCommand.Connection.Close();

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Insert, true)]

public virtual int Insert(string 职员号, string 姓名, string 性别, System.Nullable<System.DateTime> 出生日期, string 地址, byte[] 照片) {

if ((职员号 == null)) {

throw new System.ArgumentNullException("职员号");

}

else {

this.Adapter.InsertCommand.Parameters[0].Value = ((string)(职员号));

}

if ((姓名 == null)) {

throw new System.ArgumentNullException("姓名");

}

else {

this.Adapter.InsertCommand.Parameters[1].Value = ((string)(姓名));

}

if ((性别 == null)) {

this.Adapter.InsertCommand.Parameters[2].Value = System.DBNull.Value;

}

else {

this.Adapter.InsertCommand.Parameters[2].Value = ((string)(性别));

}

if ((出生日期.HasValue == true)) {

this.Adapter.InsertCommand.Parameters[3].Value = ((System.DateTime)(出生日期.Value));

}

else {

this.Adapter.InsertCommand.Parameters[3].Value = System.DBNull.Value;

}

if ((地址 == null)) {

this.Adapter.InsertCommand.Parameters[4].Value = System.DBNull.Value;

}

else {

this.Adapter.InsertCommand.Parameters[4].Value = ((string)(地址));

}

if ((照片 == null)) {

this.Adapter.InsertCommand.Parameters[5].Value = System.DBNull.Value;

}

else {

this.Adapter.InsertCommand.Parameters[5].Value = ((byte[])(照片));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.InsertCommand.Connection.State;

if (((this.Adapter.InsertCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.InsertCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.InsertCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.InsertCommand.Connection.Close();

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Update, true)]

public virtual int Update(string 职员号, string 姓名, string 性别, System.Nullable<System.DateTime> 出生日期, string 地址, byte[] 照片, string Original\_职员号, string Original\_姓名, string Original\_性别, System.Nullable<System.DateTime> Original\_出生日期, string Original\_地址) {

if ((职员号 == null)) {

throw new System.ArgumentNullException("职员号");

}

else {

this.Adapter.UpdateCommand.Parameters[0].Value = ((string)(职员号));

}

if ((姓名 == null)) {

throw new System.ArgumentNullException("姓名");

}

else {

this.Adapter.UpdateCommand.Parameters[1].Value = ((string)(姓名));

}

if ((性别 == null)) {

this.Adapter.UpdateCommand.Parameters[2].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[2].Value = ((string)(性别));

}

if ((出生日期.HasValue == true)) {

this.Adapter.UpdateCommand.Parameters[3].Value = ((System.DateTime)(出生日期.Value));

}

else {

this.Adapter.UpdateCommand.Parameters[3].Value = System.DBNull.Value;

}

if ((地址 == null)) {

this.Adapter.UpdateCommand.Parameters[4].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[4].Value = ((string)(地址));

}

if ((照片 == null)) {

this.Adapter.UpdateCommand.Parameters[5].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[5].Value = ((byte[])(照片));

}

if ((Original\_职员号 == null)) {

throw new System.ArgumentNullException("Original\_职员号");

}

else {

this.Adapter.UpdateCommand.Parameters[6].Value = ((string)(Original\_职员号));

}

if ((Original\_姓名 == null)) {

throw new System.ArgumentNullException("Original\_姓名");

}

else {

this.Adapter.UpdateCommand.Parameters[7].Value = ((string)(Original\_姓名));

}

if ((Original\_性别 == null)) {

this.Adapter.UpdateCommand.Parameters[8].Value = ((object)(1));

this.Adapter.UpdateCommand.Parameters[9].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[8].Value = ((object)(0));

this.Adapter.UpdateCommand.Parameters[9].Value = ((string)(Original\_性别));

}

if ((Original\_出生日期.HasValue == true)) {

this.Adapter.UpdateCommand.Parameters[10].Value = ((object)(0));

this.Adapter.UpdateCommand.Parameters[11].Value = ((System.DateTime)(Original\_出生日期.Value));

}

else {

this.Adapter.UpdateCommand.Parameters[10].Value = ((object)(1));

this.Adapter.UpdateCommand.Parameters[11].Value = System.DBNull.Value;

}

if ((Original\_地址 == null)) {

this.Adapter.UpdateCommand.Parameters[12].Value = ((object)(1));

this.Adapter.UpdateCommand.Parameters[13].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[12].Value = ((object)(0));

this.Adapter.UpdateCommand.Parameters[13].Value = ((string)(Original\_地址));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.UpdateCommand.Connection.State;

if (((this.Adapter.UpdateCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.UpdateCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.UpdateCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.UpdateCommand.Connection.Close();

}

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.ComponentModel.DataObjectAttribute(true)]

[System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +

", Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class OperatorTableAdapter : System.ComponentModel.Component {

private System.Data.SqlClient.SqlDataAdapter \_adapter;

private System.Data.SqlClient.SqlConnection \_connection;

private System.Data.SqlClient.SqlCommand[] \_commandCollection;

private bool \_clearBeforeFill;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public OperatorTableAdapter() {

this.ClearBeforeFill = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private System.Data.SqlClient.SqlDataAdapter Adapter {

get {

if ((this.\_adapter == null)) {

this.InitAdapter();

}

return this.\_adapter;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal System.Data.SqlClient.SqlConnection Connection {

get {

if ((this.\_connection == null)) {

this.InitConnection();

}

return this.\_connection;

}

set {

this.\_connection = value;

if ((this.Adapter.InsertCommand != null)) {

this.Adapter.InsertCommand.Connection = value;

}

if ((this.Adapter.DeleteCommand != null)) {

this.Adapter.DeleteCommand.Connection = value;

}

if ((this.Adapter.UpdateCommand != null)) {

this.Adapter.UpdateCommand.Connection = value;

}

for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {

if ((this.CommandCollection[i] != null)) {

((System.Data.SqlClient.SqlCommand)(this.CommandCollection[i])).Connection = value;

}

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected System.Data.SqlClient.SqlCommand[] CommandCollection {

get {

if ((this.\_commandCollection == null)) {

this.InitCommandCollection();

}

return this.\_commandCollection;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool ClearBeforeFill {

get {

return this.\_clearBeforeFill;

}

set {

this.\_clearBeforeFill = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitAdapter() {

this.\_adapter = new System.Data.SqlClient.SqlDataAdapter();

System.Data.Common.DataTableMapping tableMapping = new System.Data.Common.DataTableMapping();

tableMapping.SourceTable = "Table";

tableMapping.DataSetTable = "Operator";

tableMapping.ColumnMappings.Add("username", "username");

tableMapping.ColumnMappings.Add("password", "password");

this.\_adapter.TableMappings.Add(tableMapping);

this.\_adapter.DeleteCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.DeleteCommand.Connection = this.Connection;

this.\_adapter.DeleteCommand.CommandText = "DELETE FROM [dbo].[Operator] WHERE (([username] = @Original\_username) AND ((@IsNu" +

"ll\_password = 1 AND [password] IS NULL) OR ([password] = @Original\_password)))";

this.\_adapter.DeleteCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_username", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "username", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_password", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.DeleteCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_password", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.InsertCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.InsertCommand.Connection = this.Connection;

this.\_adapter.InsertCommand.CommandText = "INSERT INTO [dbo].[Operator] ([username], [password]) VALUES (@username, @passwor" +

"d);\r\nSELECT username, password FROM Operator WHERE (username = @username)";

this.\_adapter.InsertCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@username", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "username", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.InsertCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@password", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand = new System.Data.SqlClient.SqlCommand();

this.\_adapter.UpdateCommand.Connection = this.Connection;

this.\_adapter.UpdateCommand.CommandText = @"UPDATE [dbo].[Operator] SET [username] = @username, [password] = @password WHERE (([username] = @Original\_username) AND ((@IsNull\_password = 1 AND [password] IS NULL) OR ([password] = @Original\_password)));

SELECT username, password FROM Operator WHERE (username = @username)";

this.\_adapter.UpdateCommand.CommandType = System.Data.CommandType.Text;

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@username", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "username", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@password", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_username", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "username", System.Data.DataRowVersion.Original, false, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@IsNull\_password", System.Data.SqlDbType.Int, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Original, true, null, "", "", ""));

this.\_adapter.UpdateCommand.Parameters.Add(new System.Data.SqlClient.SqlParameter("@Original\_password", System.Data.SqlDbType.NChar, 0, System.Data.ParameterDirection.Input, 0, 0, "password", System.Data.DataRowVersion.Original, false, null, "", "", ""));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitConnection() {

this.\_connection = new System.Data.SqlClient.SqlConnection();

this.\_connection.ConnectionString = global::BankEmployee.Properties.Settings.Default.EmployeeConnectionString;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitCommandCollection() {

this.\_commandCollection = new System.Data.SqlClient.SqlCommand[1];

this.\_commandCollection[0] = new System.Data.SqlClient.SqlCommand();

this.\_commandCollection[0].Connection = this.Connection;

this.\_commandCollection[0].CommandText = "SELECT username, password FROM dbo.Operator";

this.\_commandCollection[0].CommandType = System.Data.CommandType.Text;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Fill, true)]

public virtual int Fill(EmployeeDataSet.OperatorDataTable dataTable) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((this.ClearBeforeFill == true)) {

dataTable.Clear();

}

int returnValue = this.Adapter.Fill(dataTable);

return returnValue;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, true)]

public virtual EmployeeDataSet.OperatorDataTable GetData() {

this.Adapter.SelectCommand = this.CommandCollection[0];

EmployeeDataSet.OperatorDataTable dataTable = new EmployeeDataSet.OperatorDataTable();

this.Adapter.Fill(dataTable);

return dataTable;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(EmployeeDataSet.OperatorDataTable dataTable) {

return this.Adapter.Update(dataTable);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(EmployeeDataSet dataSet) {

return this.Adapter.Update(dataSet, "Operator");

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(System.Data.DataRow dataRow) {

return this.Adapter.Update(new System.Data.DataRow[] {

dataRow});

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public virtual int Update(System.Data.DataRow[] dataRows) {

return this.Adapter.Update(dataRows);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Delete, true)]

public virtual int Delete(string Original\_username, string Original\_password) {

if ((Original\_username == null)) {

throw new System.ArgumentNullException("Original\_username");

}

else {

this.Adapter.DeleteCommand.Parameters[0].Value = ((string)(Original\_username));

}

if ((Original\_password == null)) {

this.Adapter.DeleteCommand.Parameters[1].Value = ((object)(1));

this.Adapter.DeleteCommand.Parameters[2].Value = System.DBNull.Value;

}

else {

this.Adapter.DeleteCommand.Parameters[1].Value = ((object)(0));

this.Adapter.DeleteCommand.Parameters[2].Value = ((string)(Original\_password));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.DeleteCommand.Connection.State;

if (((this.Adapter.DeleteCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.DeleteCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.DeleteCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.DeleteCommand.Connection.Close();

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Insert, true)]

public virtual int Insert(string username, string password) {

if ((username == null)) {

throw new System.ArgumentNullException("username");

}

else {

this.Adapter.InsertCommand.Parameters[0].Value = ((string)(username));

}

if ((password == null)) {

this.Adapter.InsertCommand.Parameters[1].Value = System.DBNull.Value;

}

else {

this.Adapter.InsertCommand.Parameters[1].Value = ((string)(password));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.InsertCommand.Connection.State;

if (((this.Adapter.InsertCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.InsertCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.InsertCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.InsertCommand.Connection.Close();

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Update, true)]

public virtual int Update(string username, string password, string Original\_username, string Original\_password) {

if ((username == null)) {

throw new System.ArgumentNullException("username");

}

else {

this.Adapter.UpdateCommand.Parameters[0].Value = ((string)(username));

}

if ((password == null)) {

this.Adapter.UpdateCommand.Parameters[1].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[1].Value = ((string)(password));

}

if ((Original\_username == null)) {

throw new System.ArgumentNullException("Original\_username");

}

else {

this.Adapter.UpdateCommand.Parameters[2].Value = ((string)(Original\_username));

}

if ((Original\_password == null)) {

this.Adapter.UpdateCommand.Parameters[3].Value = ((object)(1));

this.Adapter.UpdateCommand.Parameters[4].Value = System.DBNull.Value;

}

else {

this.Adapter.UpdateCommand.Parameters[3].Value = ((object)(0));

this.Adapter.UpdateCommand.Parameters[4].Value = ((string)(Original\_password));

}

System.Data.ConnectionState previousConnectionState = this.Adapter.UpdateCommand.Connection.State;

if (((this.Adapter.UpdateCommand.Connection.State & System.Data.ConnectionState.Open)

!= System.Data.ConnectionState.Open)) {

this.Adapter.UpdateCommand.Connection.Open();

}

try {

int returnValue = this.Adapter.UpdateCommand.ExecuteNonQuery();

return returnValue;

}

finally {

if ((previousConnectionState == System.Data.ConnectionState.Closed)) {

this.Adapter.UpdateCommand.Connection.Close();

}

}

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.ComponentModel.DataObjectAttribute(true)]

[System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +

", Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class SelectAllUserNameFromOperatorTableAdapter : System.ComponentModel.Component {

private System.Data.SqlClient.SqlDataAdapter \_adapter;

private System.Data.SqlClient.SqlConnection \_connection;

private System.Data.SqlClient.SqlCommand[] \_commandCollection;

private bool \_clearBeforeFill;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectAllUserNameFromOperatorTableAdapter() {

this.ClearBeforeFill = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private System.Data.SqlClient.SqlDataAdapter Adapter {

get {

if ((this.\_adapter == null)) {

this.InitAdapter();

}

return this.\_adapter;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal System.Data.SqlClient.SqlConnection Connection {

get {

if ((this.\_connection == null)) {

this.InitConnection();

}

return this.\_connection;

}

set {

this.\_connection = value;

if ((this.Adapter.InsertCommand != null)) {

this.Adapter.InsertCommand.Connection = value;

}

if ((this.Adapter.DeleteCommand != null)) {

this.Adapter.DeleteCommand.Connection = value;

}

if ((this.Adapter.UpdateCommand != null)) {

this.Adapter.UpdateCommand.Connection = value;

}

for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {

if ((this.CommandCollection[i] != null)) {

((System.Data.SqlClient.SqlCommand)(this.CommandCollection[i])).Connection = value;

}

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected System.Data.SqlClient.SqlCommand[] CommandCollection {

get {

if ((this.\_commandCollection == null)) {

this.InitCommandCollection();

}

return this.\_commandCollection;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool ClearBeforeFill {

get {

return this.\_clearBeforeFill;

}

set {

this.\_clearBeforeFill = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitAdapter() {

this.\_adapter = new System.Data.SqlClient.SqlDataAdapter();

System.Data.Common.DataTableMapping tableMapping = new System.Data.Common.DataTableMapping();

tableMapping.SourceTable = "Table";

tableMapping.DataSetTable = "SelectAllUserNameFromOperator";

tableMapping.ColumnMappings.Add("username", "username");

this.\_adapter.TableMappings.Add(tableMapping);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitConnection() {

this.\_connection = new System.Data.SqlClient.SqlConnection();

this.\_connection.ConnectionString = global::BankEmployee.Properties.Settings.Default.EmployeeConnectionString;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitCommandCollection() {

this.\_commandCollection = new System.Data.SqlClient.SqlCommand[1];

this.\_commandCollection[0] = new System.Data.SqlClient.SqlCommand();

this.\_commandCollection[0].Connection = this.Connection;

this.\_commandCollection[0].CommandText = "dbo.SelectAllUserNameFromOperator";

this.\_commandCollection[0].CommandType = System.Data.CommandType.StoredProcedure;

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@RETURN\_VALUE", System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.ReturnValue, 10, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Fill, true)]

public virtual int Fill(EmployeeDataSet.SelectAllUserNameFromOperatorDataTable dataTable) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((this.ClearBeforeFill == true)) {

dataTable.Clear();

}

int returnValue = this.Adapter.Fill(dataTable);

return returnValue;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, true)]

public virtual EmployeeDataSet.SelectAllUserNameFromOperatorDataTable GetData() {

this.Adapter.SelectCommand = this.CommandCollection[0];

EmployeeDataSet.SelectAllUserNameFromOperatorDataTable dataTable = new EmployeeDataSet.SelectAllUserNameFromOperatorDataTable();

this.Adapter.Fill(dataTable);

return dataTable;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.ComponentModel.DataObjectAttribute(true)]

[System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +

", Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class SelectOperatorTableAdapter : System.ComponentModel.Component {

private System.Data.SqlClient.SqlDataAdapter \_adapter;

private System.Data.SqlClient.SqlConnection \_connection;

private System.Data.SqlClient.SqlCommand[] \_commandCollection;

private bool \_clearBeforeFill;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SelectOperatorTableAdapter() {

this.ClearBeforeFill = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private System.Data.SqlClient.SqlDataAdapter Adapter {

get {

if ((this.\_adapter == null)) {

this.InitAdapter();

}

return this.\_adapter;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal System.Data.SqlClient.SqlConnection Connection {

get {

if ((this.\_connection == null)) {

this.InitConnection();

}

return this.\_connection;

}

set {

this.\_connection = value;

if ((this.Adapter.InsertCommand != null)) {

this.Adapter.InsertCommand.Connection = value;

}

if ((this.Adapter.DeleteCommand != null)) {

this.Adapter.DeleteCommand.Connection = value;

}

if ((this.Adapter.UpdateCommand != null)) {

this.Adapter.UpdateCommand.Connection = value;

}

for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {

if ((this.CommandCollection[i] != null)) {

((System.Data.SqlClient.SqlCommand)(this.CommandCollection[i])).Connection = value;

}

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected System.Data.SqlClient.SqlCommand[] CommandCollection {

get {

if ((this.\_commandCollection == null)) {

this.InitCommandCollection();

}

return this.\_commandCollection;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool ClearBeforeFill {

get {

return this.\_clearBeforeFill;

}

set {

this.\_clearBeforeFill = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitAdapter() {

this.\_adapter = new System.Data.SqlClient.SqlDataAdapter();

System.Data.Common.DataTableMapping tableMapping = new System.Data.Common.DataTableMapping();

tableMapping.SourceTable = "Table";

tableMapping.DataSetTable = "SelectOperator";

tableMapping.ColumnMappings.Add("username", "username");

tableMapping.ColumnMappings.Add("password", "password");

this.\_adapter.TableMappings.Add(tableMapping);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitConnection() {

this.\_connection = new System.Data.SqlClient.SqlConnection();

this.\_connection.ConnectionString = global::BankEmployee.Properties.Settings.Default.EmployeeConnectionString;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitCommandCollection() {

this.\_commandCollection = new System.Data.SqlClient.SqlCommand[1];

this.\_commandCollection[0] = new System.Data.SqlClient.SqlCommand();

this.\_commandCollection[0].Connection = this.Connection;

this.\_commandCollection[0].CommandText = "dbo.SelectOperator";

this.\_commandCollection[0].CommandType = System.Data.CommandType.StoredProcedure;

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@RETURN\_VALUE", System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.ReturnValue, 10, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@username", System.Data.SqlDbType.NChar, 10, System.Data.ParameterDirection.Input, 0, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@password", System.Data.SqlDbType.NChar, 20, System.Data.ParameterDirection.Input, 0, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Fill, true)]

public virtual int Fill(EmployeeDataSet.SelectOperatorDataTable dataTable, string username, string password) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((username == null)) {

this.Adapter.SelectCommand.Parameters[1].Value = System.DBNull.Value;

}

else {

this.Adapter.SelectCommand.Parameters[1].Value = ((string)(username));

}

if ((password == null)) {

this.Adapter.SelectCommand.Parameters[2].Value = System.DBNull.Value;

}

else {

this.Adapter.SelectCommand.Parameters[2].Value = ((string)(password));

}

if ((this.ClearBeforeFill == true)) {

dataTable.Clear();

}

int returnValue = this.Adapter.Fill(dataTable);

return returnValue;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, true)]

public virtual EmployeeDataSet.SelectOperatorDataTable GetData(string username, string password) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((username == null)) {

this.Adapter.SelectCommand.Parameters[1].Value = System.DBNull.Value;

}

else {

this.Adapter.SelectCommand.Parameters[1].Value = ((string)(username));

}

if ((password == null)) {

this.Adapter.SelectCommand.Parameters[2].Value = System.DBNull.Value;

}

else {

this.Adapter.SelectCommand.Parameters[2].Value = ((string)(password));

}

EmployeeDataSet.SelectOperatorDataTable dataTable = new EmployeeDataSet.SelectOperatorDataTable();

this.Adapter.Fill(dataTable);

return dataTable;

}

}

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Data.Design.TypedDataSetGenerator", "2.0.0.0")]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.ComponentModel.ToolboxItem(true)]

[System.ComponentModel.DataObjectAttribute(true)]

[System.ComponentModel.DesignerAttribute("Microsoft.VSDesigner.DataSource.Design.TableAdapterDesigner, Microsoft.VSDesigner" +

", Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a")]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

public partial class SimpleQueryTableAdapter : System.ComponentModel.Component {

private System.Data.SqlClient.SqlDataAdapter \_adapter;

private System.Data.SqlClient.SqlConnection \_connection;

private System.Data.SqlClient.SqlCommand[] \_commandCollection;

private bool \_clearBeforeFill;

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public SimpleQueryTableAdapter() {

this.ClearBeforeFill = true;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private System.Data.SqlClient.SqlDataAdapter Adapter {

get {

if ((this.\_adapter == null)) {

this.InitAdapter();

}

return this.\_adapter;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

internal System.Data.SqlClient.SqlConnection Connection {

get {

if ((this.\_connection == null)) {

this.InitConnection();

}

return this.\_connection;

}

set {

this.\_connection = value;

if ((this.Adapter.InsertCommand != null)) {

this.Adapter.InsertCommand.Connection = value;

}

if ((this.Adapter.DeleteCommand != null)) {

this.Adapter.DeleteCommand.Connection = value;

}

if ((this.Adapter.UpdateCommand != null)) {

this.Adapter.UpdateCommand.Connection = value;

}

for (int i = 0; (i < this.CommandCollection.Length); i = (i + 1)) {

if ((this.CommandCollection[i] != null)) {

((System.Data.SqlClient.SqlCommand)(this.CommandCollection[i])).Connection = value;

}

}

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

protected System.Data.SqlClient.SqlCommand[] CommandCollection {

get {

if ((this.\_commandCollection == null)) {

this.InitCommandCollection();

}

return this.\_commandCollection;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

public bool ClearBeforeFill {

get {

return this.\_clearBeforeFill;

}

set {

this.\_clearBeforeFill = value;

}

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitAdapter() {

this.\_adapter = new System.Data.SqlClient.SqlDataAdapter();

System.Data.Common.DataTableMapping tableMapping = new System.Data.Common.DataTableMapping();

tableMapping.SourceTable = "Table";

tableMapping.DataSetTable = "SimpleQuery";

tableMapping.ColumnMappings.Add("@maleNumber", "@maleNumber");

this.\_adapter.TableMappings.Add(tableMapping);

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitConnection() {

this.\_connection = new System.Data.SqlClient.SqlConnection();

this.\_connection.ConnectionString = global::BankEmployee.Properties.Settings.Default.EmployeeConnectionString;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

private void InitCommandCollection() {

this.\_commandCollection = new System.Data.SqlClient.SqlCommand[1];

this.\_commandCollection[0] = new System.Data.SqlClient.SqlCommand();

this.\_commandCollection[0].Connection = this.Connection;

this.\_commandCollection[0].CommandText = "dbo.SimpleQuery";

this.\_commandCollection[0].CommandType = System.Data.CommandType.StoredProcedure;

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@RETURN\_VALUE", System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.ReturnValue, 10, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@maleNumber", System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.InputOutput, 10, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

this.\_commandCollection[0].Parameters.Add(new System.Data.SqlClient.SqlParameter("@femaleNumber", System.Data.SqlDbType.Int, 4, System.Data.ParameterDirection.InputOutput, 10, 0, null, System.Data.DataRowVersion.Current, false, null, "", "", ""));

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Fill, true)]

public virtual int Fill(EmployeeDataSet.SimpleQueryDataTable dataTable, ref System.Nullable<int> maleNumber, ref System.Nullable<int> femaleNumber) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((maleNumber.HasValue == true)) {

this.Adapter.SelectCommand.Parameters[1].Value = ((int)(maleNumber.Value));

}

else {

this.Adapter.SelectCommand.Parameters[1].Value = System.DBNull.Value;

}

if ((femaleNumber.HasValue == true)) {

this.Adapter.SelectCommand.Parameters[2].Value = ((int)(femaleNumber.Value));

}

else {

this.Adapter.SelectCommand.Parameters[2].Value = System.DBNull.Value;

}

if ((this.ClearBeforeFill == true)) {

dataTable.Clear();

}

int returnValue = this.Adapter.Fill(dataTable);

if (((this.Adapter.SelectCommand.Parameters[1].Value == null)

|| (this.Adapter.SelectCommand.Parameters[1].Value.GetType() == typeof(System.DBNull)))) {

maleNumber = new System.Nullable<int>();

}

else {

maleNumber = new System.Nullable<int>(((int)(this.Adapter.SelectCommand.Parameters[1].Value)));

}

if (((this.Adapter.SelectCommand.Parameters[2].Value == null)

|| (this.Adapter.SelectCommand.Parameters[2].Value.GetType() == typeof(System.DBNull)))) {

femaleNumber = new System.Nullable<int>();

}

else {

femaleNumber = new System.Nullable<int>(((int)(this.Adapter.SelectCommand.Parameters[2].Value)));

}

return returnValue;

}

[System.Diagnostics.DebuggerNonUserCodeAttribute()]

[System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")]

[System.ComponentModel.DataObjectMethodAttribute(System.ComponentModel.DataObjectMethodType.Select, true)]

public virtual EmployeeDataSet.SimpleQueryDataTable GetData(ref System.Nullable<int> maleNumber, ref System.Nullable<int> femaleNumber) {

this.Adapter.SelectCommand = this.CommandCollection[0];

if ((maleNumber.HasValue == true)) {

this.Adapter.SelectCommand.Parameters[1].Value = ((int)(maleNumber.Value));

}

else {

this.Adapter.SelectCommand.Parameters[1].Value = System.DBNull.Value;

}

if ((femaleNumber.HasValue == true)) {

this.Adapter.SelectCommand.Parameters[2].Value = ((int)(femaleNumber.Value));

}

else {

this.Adapter.SelectCommand.Parameters[2].Value = System.DBNull.Value;

}

EmployeeDataSet.SimpleQueryDataTable dataTable = new EmployeeDataSet.SimpleQueryDataTable();

this.Adapter.Fill(dataTable);

if (((this.Adapter.SelectCommand.Parameters[1].Value == null)

|| (this.Adapter.SelectCommand.Parameters[1].Value.GetType() == typeof(System.DBNull)))) {

maleNumber = new System.Nullable<int>();

}

else {

maleNumber = new System.Nullable<int>(((int)(this.Adapter.SelectCommand.Parameters[1].Value)));

}

if (((this.Adapter.SelectCommand.Parameters[2].Value == null)

|| (this.Adapter.SelectCommand.Parameters[2].Value.GetType() == typeof(System.DBNull)))) {

femaleNumber = new System.Nullable<int>();

}

else {

femaleNumber = new System.Nullable<int>(((int)(this.Adapter.SelectCommand.Parameters[2].Value)));

}

return dataTable;

}

}

}

## 四、问题及解答

(1). 写出你认为有必要解释的关键步骤或代码。

见上述代码。

(2). 写出实验中遇到的问题及解决方法。

有时候FinControl（）函数找不到控件。

解决方法：重新编译一下

## 五、实验小结

通过此次实验基本掌握了绑定DataGrindView以及其他控件到数据源的方法，了解了数据库中有关图片的显示和存取。