

ТЕХНОЛОГИЯ КОНСТРУИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

При конструировании программ необходимо обеспечить пользователя продуманной технологией ввода данных: данные, вводимые пользователем, лучше оформлять в виде доступных списков, тогда товар «огурцы зеленые» не станет еще одним товаром «зеленые огурцы».

Интерфейс к базе данных: формы для отображения связанных таблиц

Рассмотрим форму, которая позволяет отображать данные трех таблиц Emp, Dep и Salaries (рисунок 1).

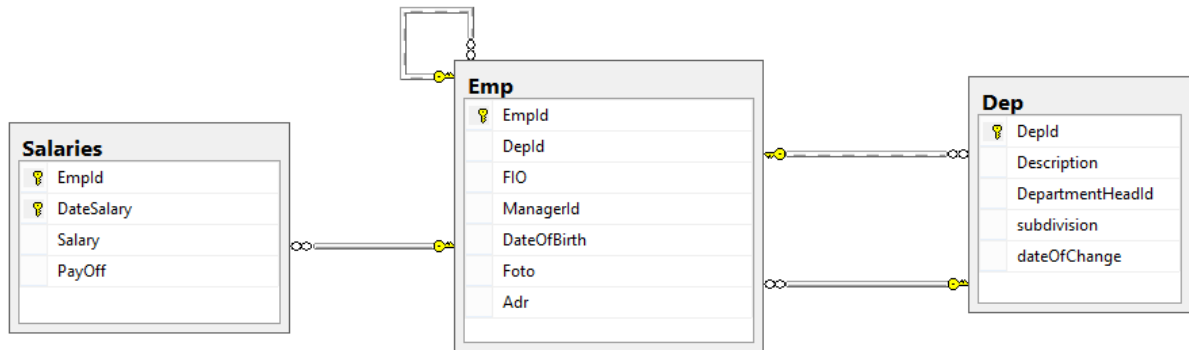


Рисунок 1 – Таблицы базы данных

Изменения данных обеспечим в двух таблицах Emp и Salaries. Данные таблицы Dep будут только отображаться, без возможности удаления, редактирования и внесения новых данных.

Создайте в вашем проекте форму, аналогичную приведенной на рисунке 2.

Сведения о сотруднике

Сотрудник
Выберите сотрудника
Петров А.А.

Редактировать сведения о сотруднике Удалить запись о сотруднике Новый сотрудник

Сведения о сотруднике

Табельный номер: 1 Отдел: Деканат

ФИО: Петров А.А.

Дата рождения: 14 января 1967 г.

Выплаты

	DateSalary	Salary	PayOff
▶	29.01.2018	10 000,00 Р	✓
	16.02.2018	20 000,00 Р	✓
	01.03.2018	10 000,00 Р	✓
<	12.03.2018	20 000,00 Р	✓

Добавить изменения Отменить

Рисунок 2 - Форма для редактирования информации о сотруднике

Добавим необходимые компоненты для обеспечения соединения с базой данных, выполнения запросов, хранения результатов выборки и выполнения синхронизации отображаемых в форме данных.

```

SqlConnection cnn;
DataSet ds = new DataSet();
DataView dvEmployees = new DataView();
DataView dvSalaries = new DataView();
SqlDataAdapter daEmployees = new SqlDataAdapter();
SqlDataAdapter daSalary = new SqlDataAdapter();
SqlDataAdapter daDepartments;

```

В соответствии с задачей, данные из таблицы Emp в форме можно просматривать, удалять, редактировать, добавлять новые записи. Ключевой столбец таблицы EmpId имеет автозаполнение. Программирование адаптера данных для работы с таблицей Emp выглядит так:

```

cnn = new SqlConnection(ConfigurationManager.ConnectionStrings
    ["Employees"].ConnectionString);
//Команда SELECT для извлечения записей о сотрудниках
daEmployees.SelectCommand = new SqlCommand("SELECT * from Emp",cnn);

//Команда Insert для вставки новых строк в Employees
daEmployees.InsertCommand = new SqlCommand("INSERT INTO Emp (DepId,FIO,
    DateOfBirth) values(@DepId,@FIO,@DateOfBirth)", cnn);
daEmployees.InsertCommand.Parameters.Add("@DepId", SqlDbType.Int, 4,
    "DepId");
daEmployees.InsertCommand.Parameters.Add("@FIO", SqlDbType.VarChar, 50,
    "FIO");
daEmployees.InsertCommand.Parameters.Add("@DateOfBirth", SqlDbType.DateTime,
    8, "DateOfBirth");

//Команда UPDATE для модификации существующих строк в таблице Employees
daEmployees.UpdateCommand = new SqlCommand("UPDATE Emp Set DepId=@DepId,
    FIO=@FIO, DateOfBirth=@DateOfBirth WHERE EmpId=@EmpId",cnn);
daEmployees.UpdateCommand.Parameters.Add("@EmpId", SqlDbType.Int, 4,
    "EmpId");
daEmployees.UpdateCommand.Parameters.Add("@DepId", SqlDbType.Int, 4,
    "DepId");
daEmployees.UpdateCommand.Parameters.Add("@FIO", SqlDbType.VarChar, 50,
    "FIO");
daEmployees.UpdateCommand.Parameters.Add("@DateOfBirth", SqlDbType.DateTime,
    8, "DateOfBirth");

//Команда DELETE для удаления существующих строк из таблицы Employees
daEmployees.DeleteCommand = new SqlCommand("DELETE Salaries WHERE
    EmpId=@EmpId; Delete Emp WHERE EmpId=@EmpId",cnn);
daEmployees.DeleteCommand.Parameters.Add("@EmpId", SqlDbType.Int, 4,
    "EmpId");

```

Данные из таблицы Salaries можно просматривать, редактировать, удалять и добавлять новые записи. Адаптер данных для этой таблицы выглядит так:

```

//Оператор SELECT для выборки данных таблицы Salaries
daSalary.SelectCommand = new SqlCommand("SELECT * from Salaries",cnn);
//Команда INSERT для вставки данных в таблицу Salaries
daSalary.InsertCommand = new SqlCommand("INSERT INTO Salaries VALUES
(@EmpId,@DateSalary, @Salary, @PayOff)",cnn);
daSalary.InsertCommand.Parameters.Add ("@EmpId", SqlDbType.Int, 4, "EmpId");
daSalary.InsertCommand.Parameters.Add ("@DateSalary", SqlDbType.DateTime, 8,
"DateSalary");
daSalary.InsertCommand.Parameters.Add ("@Salary", SqlDbType.Money, 8, "Salary");
daSalary.InsertCommand.Parameters.Add ("@PayOff", SqlDbType.Bit, 1, "PayOff");
//Команда UPDATE для обновления информации в таблице Salaries
daSalary.UpdateCommand = new SqlCommand("UPDATE Salaries Set
DateSalary=@NewDateSalary, Salary=@Salary, PayOff=@PayOff WHERE EmpID=@EmpId and
DateSalary=@OldDateSalary",cnn);
daSalary.UpdateCommand.Parameters.Add ("@EmpId", SqlDbType.Int, 4, "EmpId");
daSalary.UpdateCommand.Parameters.Add ("@Salary", SqlDbType.Money, 8, "Salary");
daSalary.UpdateCommand.Parameters.Add ("@NewDateSalary", SqlDbType.DateTime, 8,
"DateSalary");
daSalary.UpdateCommand.Parameters.Add ("@OldDateSalary", SqlDbType.DateTime, 8,
"DateSalary");
daSalary.UpdateCommand.Parameters["@OldDateSalary"].SourceVersion =
DataRowVersion.Original;
daSalary.UpdateCommand.Parameters
["@NewDateSalary"].SourceVersion=DataRowVersion.Current;
daSalary.UpdateCommand.Parameters.Add ("@PayOff", SqlDbType.Bit, 1, "PayOff");
//Оператор DELETE, предназначенный для удаления данных из таблицы Salaries
daSalary.DeleteCommand = new SqlCommand("DELETE Salaries WHERE EmpId=@EmpId and
DateSalary=@DateSalary",cnn);
daSalary.DeleteCommand.Parameters.Add ("@EmpId", SqlDbType.Int, 4, "EmpId");
daSalary.DeleteCommand.Parameters.Add ("@DateSalary", SqlDbType.DateTime, 8,
"DateSalary");

```

Данные из таблицы Dep можно только просматривать:

```

//Departments
daDepartments = new SqlDataAdapter("select DepId, Description from Dep",cnn);

```

Получение данных и размещение в DataSet:

```

//Выполним загрузку данных
daEmployees.Fill(ds,"Emp");
daSalary.Fill(ds,"Sal");
daDepartments.Fill(ds, "Dep");

```

```

//Создадим представление для таблицы Emp
dvEmployees.Table = ds.Tables["Emp"];

```

Теперь можно выполнить связь полученных наборов данных с элементами интерфейса.

```

//Свяжем столбцы EmpId и FIO таблицы Emp с comboBox1
comboBox1.DataSource = ds.Tables["Emp"];
comboBox1.ValueMember = "EmpId";
comboBox1.DisplayMember = "FIO";

```

Сотрудник

Выберите сотрудника

```
//Отображение отдела в comboBox2
comboBox2.DataSource = ds.Tables["Dep"];
comboBox2.ValueMember = "DepId";
comboBox2.DisplayMember = "Description";
```

Отдел

```
//Отображение деталей информации о сотруднике
RefreshEmp();
```

Сведения о сотруднике

Табельный номер

Отдел

ФИО

Дата рождения

```
//Представление для таблицы Salaries
dvSalaries.Table = ds.Tables["Sal"];
dvSalaries.RowFilter = "EmpId=-1";
dataGridView1.DataSource = dvSalaries;
//Форматирование столбцов dataGridView для отображения денег и значений типа NULL
dataGridView1.Columns["Salary"].DefaultCellStyle.Format = "c";
dataGridView1.DefaultCellStyle.NullValue = "";
dataGridView1.Columns[0].Visible = false;
groupBox2.Enabled = false;
```

Выплаты

Напишем метод, позволяющий отобразить текущие данные о сотруднике:

```
//Метод отображения данных о сотруднике
```

ссылка 4

```
private void RefreshEmp()
{
    textBox1.Text = dvEmployees[0]["EmpId"].ToString();
    textBox2.Text = dvEmployees[0]["FIO"].ToString();
    dateTimePicker1.Value = Convert.ToDateTime(dvEmployees[0]["DateOfBirth"]);
    comboBox2.SelectedValue = dvEmployees[0]["DepId"];
}
```

Кнопка «Редактировать» позволяет сделать доступной нижнюю часть формы. Кнопка «Отмена» отменяет режим редактирования (нижняя часть формы становится недоступной).

//Кнопка Редактировать

ссылка 1

```
private void button2_Click(object sender, EventArgs e)
{
    dvSalaries.RowFilter = String.Format("EmpId={0}", comboBox1.SelectedValue);
    dvSalaries.Table.Columns["EmpId"].DefaultValue = comboBox1.SelectedValue;
    dvEmployees.RowFilter = String.Format("EmpId={0}", comboBox1.SelectedValue);
    groupBox2.Enabled = true;
    groupBox1.Enabled = false;
    RefreshEmp();
}
```

//Кнопка Отмена

ссылка 1

```
private void button4_Click(object sender, EventArgs e)
{
    ds.RejectChanges();
    groupBox1.Enabled = true;
    groupBox2.Enabled = false;
    dvEmployees.RowFilter = String.Format("EmpId={0}", comboBox1.SelectedValue);
    RefreshEmp();
    dvSalaries.RowFilter = "EmpId=-1";
}
```

Кнопка «Добавить» нижней части формы позволяет сохранить внесенные изменения и вернуться в верхнюю панель навигации.

//Кнопка Добавить

ссылка 1

```
private void button5_Click(object sender, EventArgs e)
{
    //Обновление информации
    dvEmployees[0]["EmpId"] = Convert.ToInt32(textBox1.Text);
    dvEmployees[0]["FIO"] = textBox2.Text;
    dvEmployees[0]["DateOfBirth"] = dateTimePicker1.Value;
    dvEmployees[0]["DepId"] = Convert.ToInt32(comboBox2.SelectedValue);
    daSalary.Update(ds.Tables["Sal"]);
    daEmployees.Update(ds.Tables["Emp"]);
    //Сделаем поля нижней части формы недоступными для редактирования,
    //откроем доступ к панели навигации
    groupBox1.Enabled = true;
    groupBox2.Enabled = false;
    dvSalaries.RowFilter = "EmpId=-1";
}
```

Удаление записи о текущем сотруднике выполняется кнопкой «Удалить».

//Кнопка Удалить

ссылка 1

```
private void button1_Click(object sender, EventArgs e)
{
    //Текущая строка
    dvEmployees.RowStateFilter = DataRowState.CurrentRows;
    dvEmployees.RowFilter = String.Format("EmpId={0}", comboBox1.SelectedValue);
    //Удалим текущую запись
    dvEmployees[0].Delete();
    //Обновим информацию
    daEmployees.Update(ds.Tables["Emp"]);
}
```

Для добавления записи о новом сотруднике можно выполнить следующие действия:

//Новый сотрудник

ссылка 1

```
private void button3_Click(object sender, EventArgs e)
{
    DataRow dr = ds.Tables["Emp"].NewRow();
    comboBox2.SelectedIndex = 1;
    dr["DepId"] = Convert.ToInt32(comboBox2.SelectedValue);
    dr["FIO"] = "Новый сотрудник";
    dr["DateOfBirth"] = dateTimePicker1.Value;
    ds.Tables["Emp"].Rows.Add(dr);
    daEmployees.Update(ds.Tables["Emp"]);
    ds.Tables["Emp"].Clear();
    daEmployees.Fill(ds, "Emp");
    dvEmployees.RowFilter = "EmpId=MAX(EmpId)";
    comboBox1.SelectedValue = dvEmployees[0]["EmpId"];
    dvSalaries.Table.Columns["EmpId"].DefaultValue = comboBox1.SelectedValue;
    groupBox1.Enabled = true;
    groupBox2.Enabled = false;
    RefreshEmp();
}
```

Событие «Закрытие формы» можно использовать для сохранения всех изменений и передачи их в базу данных.

//При закрытии формы обновление данных в БД

ссылка 1

```
private void Form15_FormClosing(object sender, FormClosingEventArgs e)
{
    //Обновление информации в таблицах Employees и Salary
    daSalary.Update(ds.Tables["Sal"]);
    daEmployees.Update(ds.Tables["Emp"]);
}
```

Задание для самостоятельной работы

Разработайте приложение с возможностью обновления, удаления и вставки новых данных для нескольких таблиц базы данных.

Выполните тестирование всех возможных ситуаций при использовании приложения: удаление, вставку, изменение записей.