

## ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Встроенные возможности DataSet позволяют производить некоторые операции с xml-документами, в частности, сохранять данные из DataSet в xml и, наоборот, загружать из xml в DataSet. Подобные возможности позволяют при необходимости использовать DataSet в качестве посредника между базой данных и xml-файлами.

### Основные принципы записи данных в xml формат

Данные, хранящиеся в DataSet независимо от способа записи, и схемы этих данных, можно прочитать из DataSet и сохранить в xml-формате (с xml-схемой или без нее).

Для изучения методов работы с данными в xml-формате спроектируйте интерфейс формы, например, такой, как показано на рисунке 1. Форма позволяет отображать связанные записи из 2-х таблиц: Отделы и Сотрудники. На рисунке 2 показан результат работы с формой:

- 1) загрузка данных из DataSet в dataGrid(ы);
- 2) чтение данных из DataSet в файл xml со схемой или без нее;
- 3) очистка данных в DataSet и dataGrid(ax);
- 4) чтение xml файла и запись в DataSet с отображением в dataGrid(ax);
- 5) чтение данных из базы данных в формате xml и отображение в richTextBox.

Рисунок 1 – Конструирование формы для исследования возможностей работы DataSet с данными в xml-формате

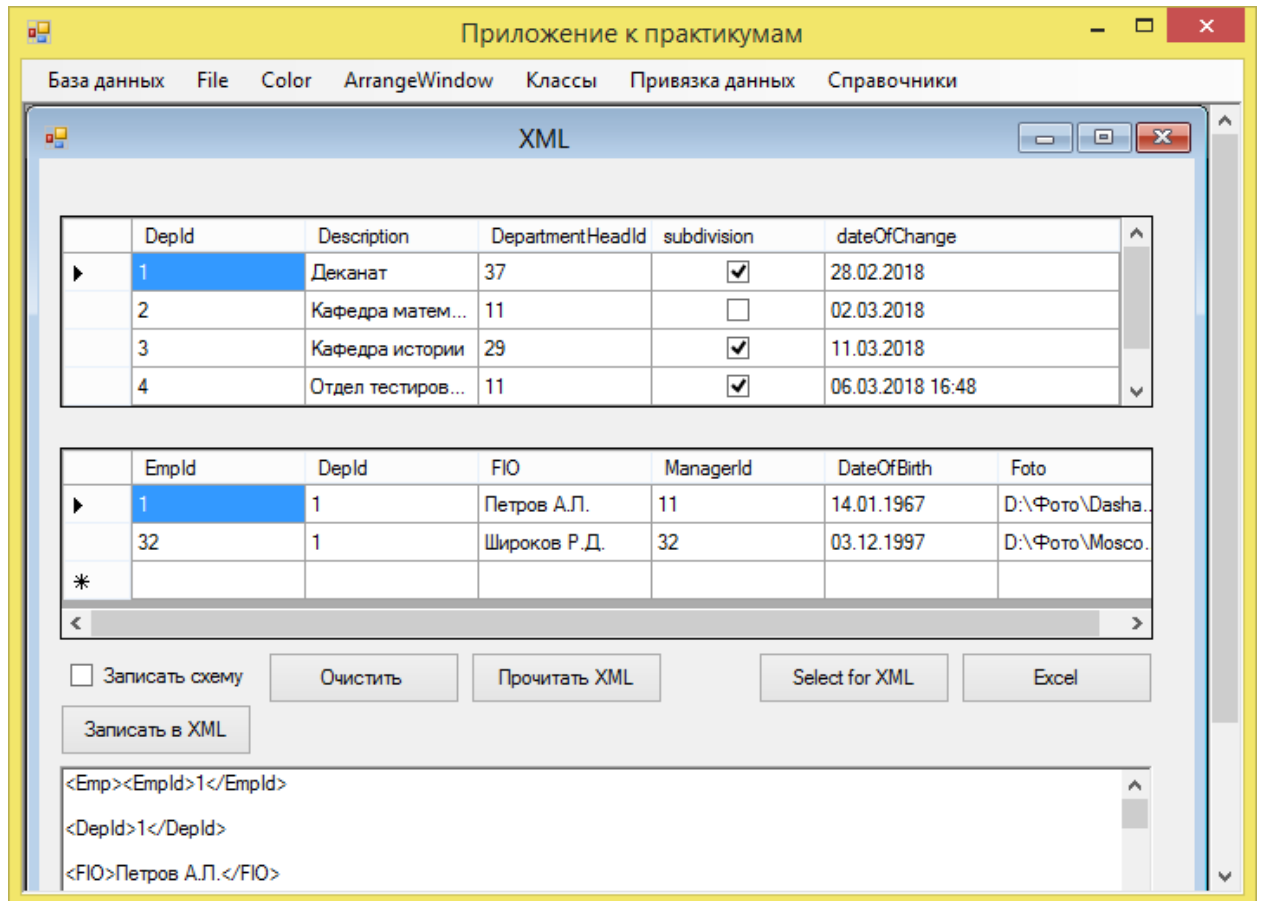


Рисунок 2 – Результаты чтения DataSet

Для записи реляционных данных в формат xml необходимо использовать методы объекта DataSet:

- метод WriteXml() позволяет преобразовать реляционные данные в xml формат,
- метод WriteXmlSchema() записывает схему отношений объектов, размещенных в DataSet.

```
//Запись объекта DataSet в файл xml
private void button1_Click(object sender, EventArgs e)
{
    saveFileDialog1.Title = "Запись файла в формате xml";
    if (saveFileDialog1.ShowDialog() != DialogResult.Cancel)
    {
        ds.WriteXml(saveFileDialog1.FileName+".xml");
    }
    if (checkBox1.Checked) //Подтверждение записи схемы
    {
        saveFileDialog1.Title = "Запись схемы отношений xsd";

        if (saveFileDialog1.ShowDialog() != DialogResult.Cancel)
        {
            ds.WriteXmlSchema(saveFileDialog1.FileName+".xsd");
        }
    }
}
```

На рисунке 3 показан файл Emp.xml, который создается в результате записи содержимого DataSet в xml -файл.

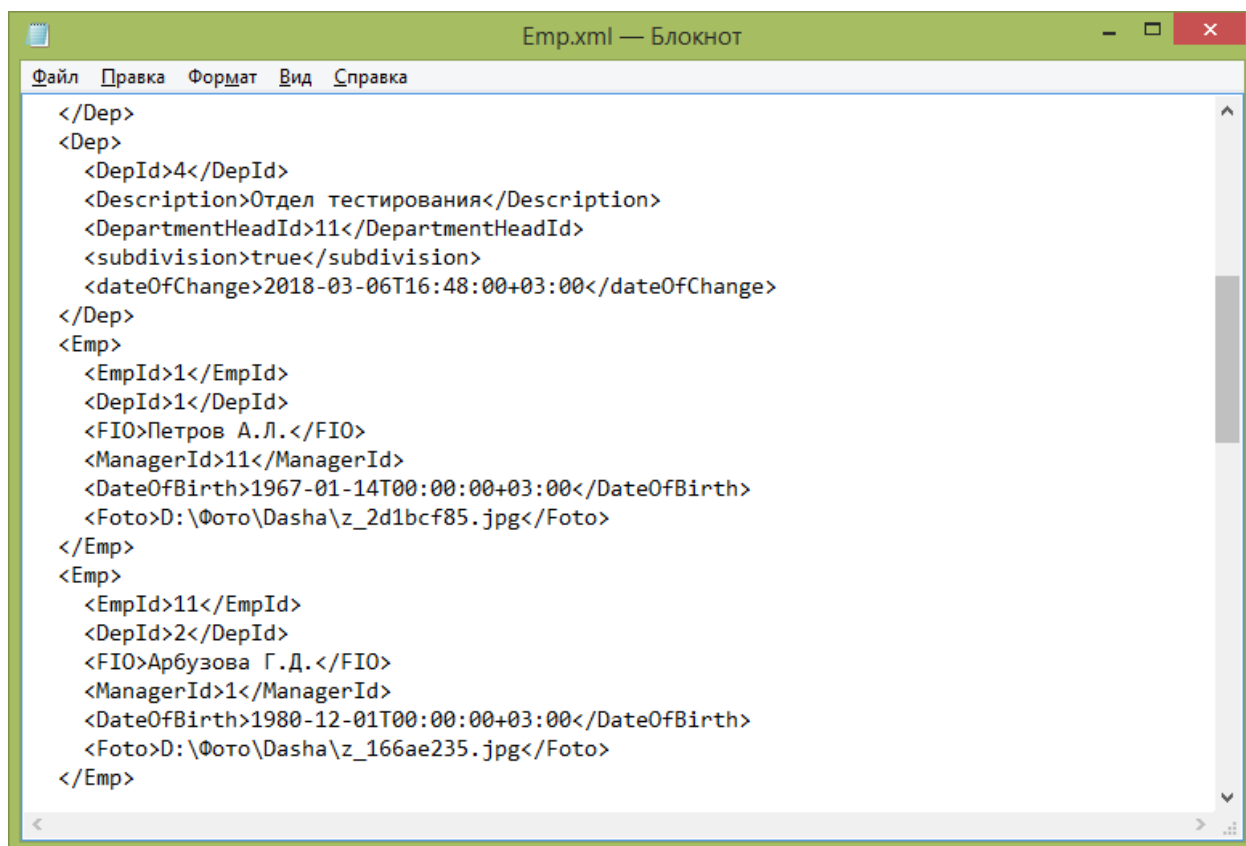


Рисунок 3 – Запись данных из DataSet в xml-файл

Обратите внимание, что xml-документ содержит записи об отделах, а ниже записи о сотрудниках, не смотря на то, что между таблицами в DataSet было установлено отношение, которое позволяет отображать связанные записи (рисунок 2).

Схема была сохранена в файл Emp.xsd. На рисунке 4 приведена схема двух связанных таблиц из файла Emp.xsd.

Для вложения дочерних записей из таблицы Emp в родительские строки таблицы Dep нужно указать значение **true** для свойства **Nested** объекта-отношения Relation, т.е. явно указать вложенность данных.

На рисунке 5 показан файл xml с вложением дочерних записей в соответствующие им родительские записи.

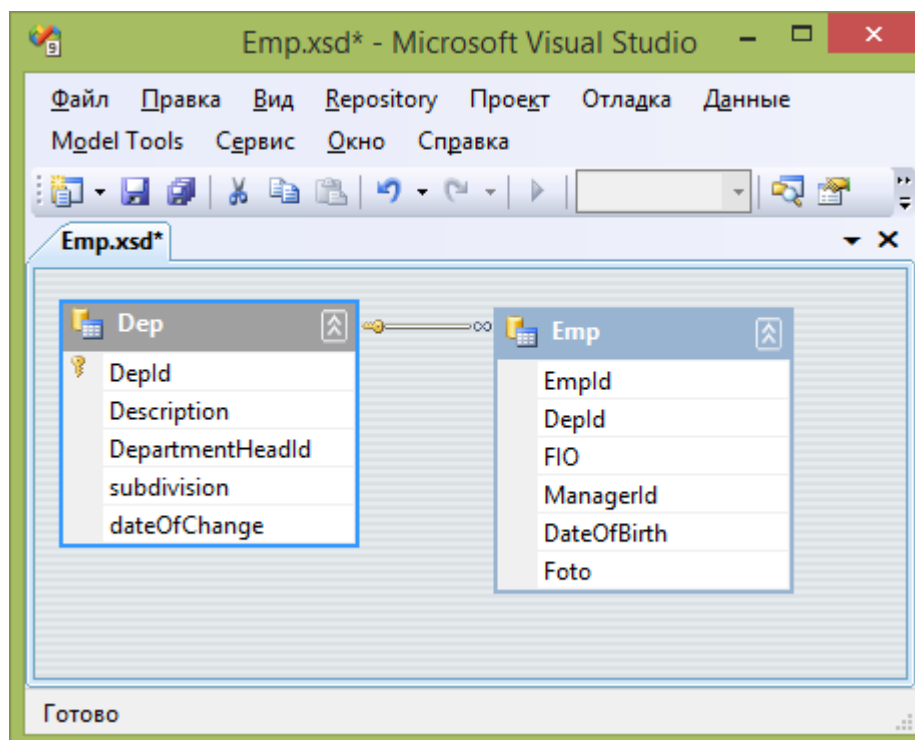


Рисунок 4 – Схема отношений между наборами данных

```

Emp.xml — Блокнот
Файл  Правка  Формат  Вид  Справка
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Dep>
    <DepId>1</DepId>
    <Description>Деканат</Description>
    <DepartmentHeadId>37</DepartmentHeadId>
    <subdivision>true</subdivision>
    <dateOfChange>2018-02-28T00:00:00+03:00</dateOfChange>
  <Emp>
    <EmpId>1</EmpId>
    <DepId>1</DepId>
    <FIO>Петров А.Л.</FIO>
    <ManagerId>11</ManagerId>
    <DateOfBirth>1967-01-14T00:00:00+03:00</DateOfBirth>
    <Foto>D:\Фото\Dasha\z_2d1bcf85.jpg</Foto>
  </Emp>
  <Emp>
    <EmpId>32</EmpId>
    <DepId>1</DepId>
    <FIO>Широков Р.Д.</FIO>
    <ManagerId>32</ManagerId>
    <DateOfBirth>1997-12-03T00:00:00+03:00</DateOfBirth>
    <Foto>D:\Фото\Moscow2015\SNV36271.JPG</Foto>
  </Emp>
</NewDataSet>

```

Рисунок 5 – Иерархия в данных

## Извлечение и обработка данных из СУБД SQL Server в формате XML

Для извлечения и доступа к xml -данным из СУБД SQL Server можно использовать метод ExecuteXmlReader() объекта Command, который возвращает объект XmlReader.

Для демонстрации применения метода ExecuteXmlReader() выполним следующие действия:

Добавьте в форму для задания 1 кнопку «Select For XML» и элемент richTextBox.

Разберите и добавьте к обработчику события щелка по кнопке следующий программный код:

```
//Выборка данных в формате XML из базы данных
private void button2_Click(object sender, EventArgs e)
{
    SqlCommand cmd = new SqlCommand("SELECT * FROM Employees FOR XML AUTO, elements", cnn);
    System.Xml.XmlReader reader;
    StringBuilder str = new StringBuilder();
    cnn.Open();
    //Выполнение SQL-команды Select с предложением FOR XML
    reader = cmd.ExecuteXmlReader();
    //Поиск и извлечение данных из узлов-элементов
    while (reader.Read())
    {
        switch (reader.NodeType)
        {
            case System.Xml.XmlNodeType.Element:
                str.Append("<" + reader.Name + ">");
                break;
            case System.Xml.XmlNodeType.EndElement:
                str.Append("</>" + reader.Name + ">" + "\n\r");
                break;
            case System.Xml.XmlNodeType.Text:
                str.Append(reader.Value);
                break;
            default:
                //Сейчас мы не будем рассматривать другие возможные XML-дескрипторы
                break;
        }
    }
    cnn.Close();
    richTextBox1.Text = str.ToString();
}
```

Предложенный к рассмотрению программный код является упрощенным примером использования метода ExecuteXmlReader(). Код примера позволяет просто отобразить данные (вместе с дескрипторами) из таблицы базы данных Emp. СУБД MS SQL Server передается SQL-команда SELECT, в которой явно указаны возвращаемые поля в виде XML-элементов:

"SELECT \* FROM Emp FOR XML AUTO, elements"

В таком случае обработка разных типов xml -узлов упрощается, потому что достаточно найти только начальный и конечный узлы, а текстовые узлы между ними будут содержать фактические данные. Для случая полного разбора xml -документа необходимо учитывать узлы всех типов.

## Экспорт данных в приложения MS Office

Элемент управления DataGridView исключительно полезен для приложений, работающих с базами данных, однако его функциональности не всегда достаточно. Часто данные необходимо экспортировать, например, в Microsoft Excel для дальнейшей обработки.

То, что MS Excel умеет отображать xml -документы, сильно упрощает задачу экспорта. Если загрузить в Excel файл xml, в котором мы сохраняли данные из формы, можно получить следующие объекты: таблицу и карту xml. Попробуйте проделать для своих xml файлов.

Но можно предложить пользователю самостоятельно не открывать Excel, а выполнить все действия по экспорту из формы (рисунок 6).

Добавьте кнопку в форму приложения и введите предложенный ниже программный код.

```

        //Необходимо добавить ссылку и пространство имен
        using Excel = Microsoft.Office.Interop.Excel;

//Экспорт в MS Excel
private void button5_Click(object sender, EventArgs e)
{
    Excel.Application ExcelApp = new Excel.Application();
    Excel.Workbook ExcelWorkBook;
    Excel.Worksheet ExcelWorkSheet;
    ExcelWorkBook = ExcelApp.Workbooks.Add();
    ExcelWorkSheet = (Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);
    for (int i = 1; i < dataGridView1.Columns.Count + 1; i++)
    {
        ExcelWorkSheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;
    }
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
    {
        for (int j = 0; j < dataGridView1.Columns.Count; j++)
        {
            ExcelWorkSheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value.ToString();
        }
    }

    ExcelApp.Visible = true;
}

```

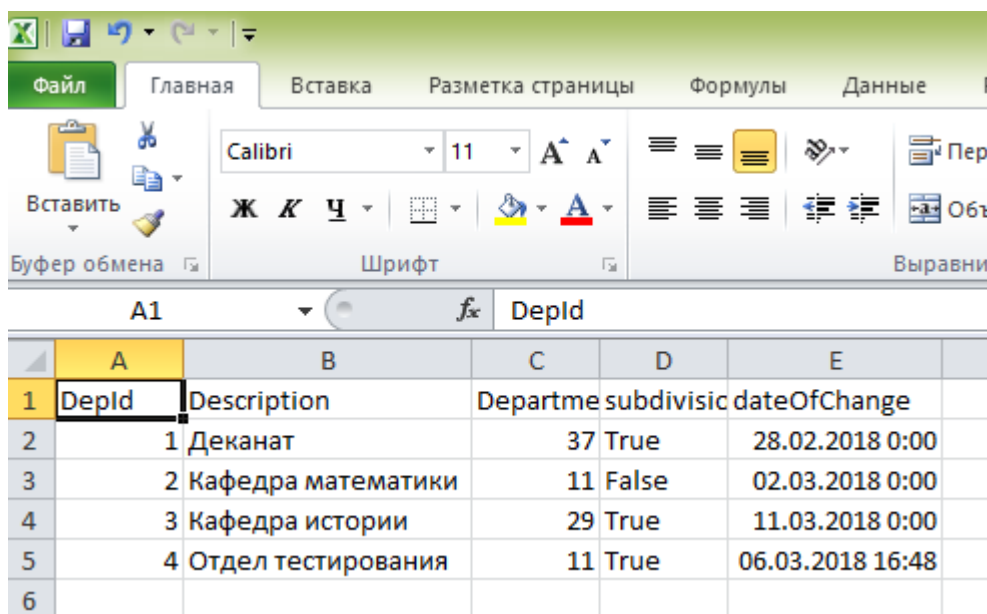


Рисунок 6 – Экспорт в Excel

## Самостоятельно. Бизнес-ситуация: подготовка XML-файлов для бизнес-партнеров

Предположим, компания обменивается информацией со своими партнерами с использованием xml формата. Необходимо организовать отправку xml-файлов с перечнем услуг, работ, покупки/продажи материалов, их стоимости, функциональности и т.д. Другими словами, необходимо сформировать коммерческое предложение в xml формате для передачи контрагентам. Состав полей в передаваемом массиве данных сформируйте самостоятельно.

Для создаваемого xml -файла существует еще одно требование: в файл необходимо включить XSD-схему с описанием всех полей как элементов.

Разработайте собственную форму для изучения возможностей dataset по чтению/записи xml-форматов. Не забудьте настроить маску для фильтрации имен файлов в диалоговых окнах:

```
openFileDialog1.Filter = "XML(*.xml)|*.xml|All files(*.*)|*.*";
```

В форме организуйте возможность обратного чтения xml-данных в dataset и отображения в интерфейсе связанных наборов данных. Для упрощения можно прочитать и отобразить тот же xml файл, который был подготовлен для партнеров. Т.е. по кнопке «Записать в XML» надо создать xml-файл, а по кнопке «Прочитать XML» открыть сохраненный xml-файл, прочитать его содержание в dataset (используйте метод ReadXML() объекта DataSet) и отобразить в форме.

Для задания 2 дополните форму возможностью читать данные в формате xml напрямую из базы данных.

Для задания 3 напишите комментарии к каждой строке программного кода, выполняющего экспортирование данных из элемента управления dataGridView в электронные таблицы MS Excel.