

ТЪРСЕНЕ И АНАЛИЗ НА ДАННИТЕ

Филтриране и сортиране на данните

- ▣ Използване на стълбове-изрази за отразяване на изчисляеми стойности
- ▣ Търсене на несъответствия между данните, съдържащи се в два обекта DataSet
- ▣ Преместване между родителски и дъщерни записи с помощта на връзка
- ▣ Локализация на данните от страна на клиента в приложения Web Forms
- ▣ Комбиниране на данни в таблици от разнородни източници
- ▣ Отразяване на стойност от вградени функции с помощта на стълбове-изрази
- ▣ Търсене на ред в локална таблица
- ▣ Търсене на ред в представяне на локална таблица
- ▣ Получаване на първите N реда от локалната таблица

- Получаване на типизирани редове от представяния

- Филтрация по стойност NULL

- Изпълнение на заявки с командата COMPUTE BY

- Използване на езика Shape за получаване на йерархично организирани данни

ВЪВЕДЕНИЕ

Обекта `DataGridView` е представяне на обекта `DataTable` с помощта на свързани данни, сортиране на записи в различен ред и филтриране по определен признак. Една таблица може да има множество представяния; всяка таблица има стандартно представяне.

Класът `DataRelation` създава връзка “родител-наследник” между две таблици от набора данни. Той обезпечават поддържането на цялостност на връзките, и той може да се използва за каскадно обновяване и изтриване. С негова помощ може да се извършва преместване между таблиците.

Двата класа, `DataTable` и `DataGridView`, предвиждат няколко способа за търсене на записи по определен критерий. Извършвайки търсене в локална таблица или представяне, може да се икономисат време и ресурси, необходими за обръщане към сървера на БД и получаване на отговор от него. Въпреки, че представянето се създава на базата на таблици, търсенето в обектите `DataTable` и `DataGridView` се изпълнява по различен начин.

В ADO.NET не е предвиден начин за сравнение на два обекта `DataSet` с идентични схеми за определяне на различията в данните.

- Стълбове-изрази. Стълбовете-изрази се изчисляват по формули, в които влизат стойности на стълбове от текущ ред или комбинация стойности от редове на същата или друга таблица, свързана с дадената. Стълба-израз не се съхранява никъде, а се изчислява отново всеки път когато се изиска неговата стойност, и може да се използва по същия начин както и обикновен ред. Например, могат да се сортират и филтрират таблици и представяния по стълб-израз.

Глобализация и локализация. Под глобализация се разбира създаването на приложения, поддържащи различни стандарти. Локализация се нарича модификация на приложение за поддръжка на определен стандарт – главно за сметка на превода на потребителския интерфейс. Пространството на имената System.Globalization съдържа класове, управляващи отразяване на стойности – пари и ги дава по начин, характерен за различните стандарти.

Сложни заявки. В ADO.NET не е предвидена възможността за получаване на първите N реда от локална таблица започвайки от стойността на даден стълб.

Понякога е необходимо в една заявка да се обединят данни от няколко източника. Специалните имена на връзките позволяват да се получат данни от разнородни източници с указване на сведенията за връзките непосредствено в SQL-оператор.

Командата COMPUTE BY дава възможност да се получат както сводни, така и подробни данни от един и същ набор редове върнат от един оператор SELECT.

Езика Shape използва DataShaping Services за OLE DB като алтернатива на командите JOIN и GROUP BY за генериране на резултиращия набор редове от йерархична структура.

Филтриране и сортиране на данните

Проблем. Налице е обект DataSet, напълнен с данни, но трябва да се работи само с

някакво подмножество на негови записи Тези записи трябва да бъдат подредени в определен ред. Необходим е способ за филтриране и сортиране на записи в обекта DataSet, без да се прави повторна заявка към източника данни.

Решение. За филтриране и сортиране на данните в обекта DataSet се използват обектите DataViewManager и DataView.

- Form.Load. Осъществява подготвителни действия: създава обекта DataSet, съдържащ таблиците Customers и Orders от БД Northwind и връзка между двете таблици. Стандартното представяне на таблицата Customers се свързва с табличен елемент в екранна форма.

- Button.Click (бутон Обнови). Прилага зададените от потребителя филтри и сортировки към представяните таблици, достъпа до които се извършва чрез обекта DataViewManager.

// Пространство на имена, променливи и константи

```
using System;
```

```
using System.Configuration;
```

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
//...
```

```
private void FilterSortForm_Load(object sender, System.EventArgs e)

{

ds = new DataSet();

SqlDataAdapter da;

// напълваме с данни таблицата Customers и я добавяме в обекта DataSet

da = new SqlDataAdapter("SELECT * FROM Customers",
ConfigurationSettings.AppSettings["Sql_ConnectString"]);

DataTable customersTable = new DataTable(CUSTOMERS_TABLE);

da.Fill(customersTable);

ds.Tables.Add(customersTable);

// попълваме с данни таблицата Orders и я добавяме в обекта DataSet

da = new SqlDataAdapter("SELECT * FROM Orders",
ConfigurationSettings.AppSettings["Sql_ConnectString"]);
```

```
DataTable orderTable = new DataTable(ORDERS_TABLE);
```

```
da.Fill(orderTable);
```

```
ds.Tables.Add(orderTable);
```

```
// създаваме връзка между таблиците
```

```
ds.Relations.Add(CUSTOMERS_ORDERS_RELATION,
```

```
ds.Tables[CUSTOMERS_TABLE].Columns[CUSTOMERID_FIELD],
```

```
ds.Tables[ORDERS_TABLE].Columns[CUSTOMERID_FIELD],
```

```
true);
```

```
// свързваме обекта DataViewManager към табличен елемент в екранна форма
```

```
dataGrid.SetDataBinding(ds.DefaultViewManager, CUSTOMERS_TABLE);
```

```
}
```

```
private void refreshButton_Click(object sender, System.EventArgs e)

{

String countryFilter = "";

if (customerCountryTextBox.Text != "")

countryFilter = "Country = " + customerCountryTextBox.Text + "";

DataManager dvm = new DataManager(ds);

// сортираме по името на лицето за контакти (ако това е поискал потребителя)

if(contactSortCheckBox.Checked)

dvm.DataViewSettings[CUSTOMERS_TABLE].Sort = CONTACTNAME_FIELD;

// филтрираме представянето на таблиците Customers по държава

dvm.DataViewSettings[CUSTOMERS_TABLE].RowFilter = countryFilter;

// филтрираме представянето н таблицата Orders по сътрудник
```

```
String employeeldFilter = "";
```

```
if (orderEmployeeIdTextBox.Text != "")
```

```
{
```

```
try
```

```
{
```

```
employeeldFilter = "EmployeeId = " + Int32.Parse(orderEmployeeIdTextBox.Text);
```

```
}
```

```
catch (FormatException)
```

```
{
```

```
orderEmployeeIdTextBox.Text = "";
```

```
}
```



```
}  
  
dvm.DataViewSettings[ORDERS_TABLE].RowFilter = employeeldFilter;  
  
// свързваме обекта DataViewManager към табличен елемент в екранна форма  
  
dataGrid.SetDataBinding(dvm, CUSTOMERS_TABLE);  
  
}
```

Обсъждане. Обекта `DataView` филтрира и сортира данните, съдържащи се в обектите `DataTable` от набора данни. Обекта `DataViewManager` може да опрости работата с множество представяния в набора данни, но използването му не е задължително. В обекта `DataViewManager` има колекция `DataViewSettingCollection`, достъпа до която се осъществява чрез свойството `DataViewSettings`. Колекцията съдържа по един обект `DataViewSetting` за всяка таблица от набора данни. Обръщането към обекта се извършва по име или по пореден номер на таблицата, за което се използва индексатора в `C#` или свойството `Item()` в `VB.NET`. Обекта `DataViewSetting` предоставя достъп до свойствата `ApplyDefaultSort`, `RowFilter`, `RowStateFilter` и `Sort` на обекта `DataView`, създаден от обекта `DataViewManager` за дадена таблица. Манипулацията на тези свойства чрез обекта `DataViewSetting` е еквивалентна на непосредствено обръщение към него чрез обекта `DataView`.