**Лабораторная работа № 4. Работа с объектом DataView**

**Цель работы:**

– получить практические навыки создания и использования объекта DataView для отображения данных.

**Теоретические сведения**

1. **Класс DataView**

Класс DataView определяет представление объекта DataTable — другими словами, представление данных в DataTable, которое может включать пользовательские настройки фильтрации и сортировки. Для конфигурирования этих настроек в DataView предусмотрены такие свойства, как Sort и RowFilter. Эти свойства позволяют выбрать данные, которые должны быть видимы в представлении. Однако они не затронут реальные данные DataTable. Например, если вы отфильтруете таблицу так, чтобы скрыть определенные строки, то эти строки останутся в DataTable, но не будут доступны через DataView.

Класс DataView позволяет показать только подмножество общего набора данных таблицы, без необходимости обрабатывать или изменять данные, если они нужны для других задач.

Каждый объект DataTable имеет ассоциированный с ним объект DataView по умолчанию, хотя допускается создавать множество объектов DataView для представления разных видов одной и той же таблицы. DataView по умолчанию представлен свойством DataTable.DefaultView.

В последующих примерах будет показано, как создавать некоторые экранные таблицы, которые отображают записи с сортировкой по различным полям, отфильтрованные в соответствии с заданным выражением.

* 1. **Сортировка с помощью DataView**

В следующем примере используется страница с тремя элементами управления GridView. Когда эта страница загружается, все элементы привязываются к одной DataTable. Однако она использует три разных представления, каждое из которых сортирует результат по разным полям:

***private void btRead\_Click(object sender, EventArgs e)***

***{***

***// Строка подключения***

***string connectionString = "Data Source=MyPC;Initial Catalog=trainingDB;Integrated Security=True";***

***SqlConnection con = new SqlConnection(connectionString);***

***string cmd = "SELECT TOP 5 id, LName, FName, EmploymentDate FROM Employees";***

***SqlDataAdapter adapter = new SqlDataAdapter(cmd, con);***

***DataSet dataset = new DataSet();***

***adapter.Fill(dataset, "Employees");***

***// Привязка исходных данных к элементу GridView1***

***dataGridView1.DataSource = dataset.Tables["Employees"];***

***// Сортировка по фамилии и привязка к элементу GridView2***

***DataView view2 = new DataView(dataset.Tables["Employees"]);***

***view2.Sort = "LName";***

***dataGridView2.DataSource = view2;***

***// Сортировка no имени и привязка к элементу GridView3***

***DataView view3 = new DataView(dataset.Tables["Employees"]);***

***view3.Sort = "FName";***

***dataGridView3.DataSource = view3;***

}

Код начинается с извлечения списка сотрудников в DataSet. На следующем шаге происходит наполнение элемента управления GridView с помощью привязки данных. Чтобы привязать первый из них, можно просто применить объект DataTable непосредственно, что заставит его использовать DataView по умолчанию и отобразить данные. Для двух других понадобится создать новый объект DataView, а затем явно использовать его свойство Sort.

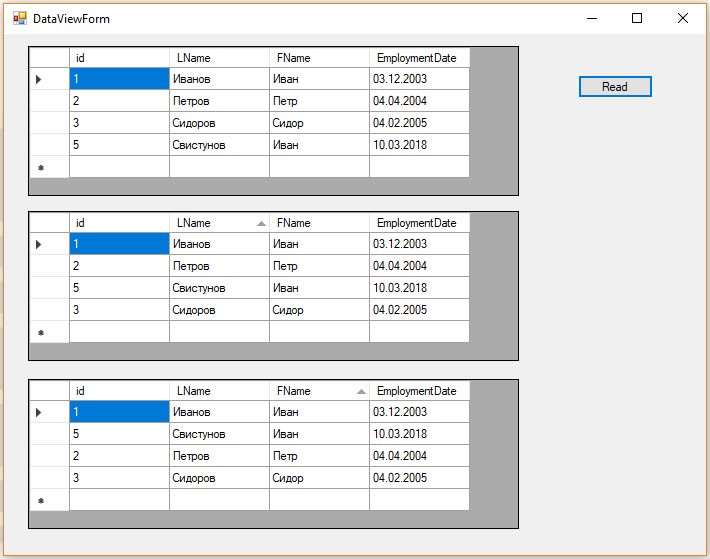
Сортировка отображаемых данных заключается просто в присваивании свойству DataView.Sort корректного выражения сортировки. В данном примере сортировка выполняется в каждом представлении по значениям единственного поля, но также можно сортировать и нескольким полям, указывая список полей, разделенных запятыми. Вот пример:

view2.Sort = "LastName, FirstName";

Сортировка осуществляется в соответствии с типом данных столбца. Числовые столбцы и столбцы дат упорядочиваются от меньших значений к большим. Строковые столбцы сортируются в алфавитно-цифровом порядке, независимо от регистра, если свойство DataTable.CaseSensitive установлено в false (по умолчанию). Столбцы, содержащие двоичные данные, сортироваться не могут. С помощью атрибутов ASC или DESC сортировка производится в возрастающем или убывающем порядке:

view2.Sort = "LastName ASC, FirstName DESC";

Внешний вид формы показан ниже



* 1. **Фильтрация с помощью DataView**

Элемент DataView можно также использовать для применения пользовательской фильтрации, чтобы отображать на экране только определенные строки. Для обеспечения этой возможности служит свойство RowFilter. Свойство RowFilter действует подобно конструкции WHERE в SQL-запросе. Используя его, результат можно ограничить с помощью логических операций (таких как <, > и =) и широкого диапазона критериев.

В таблице ниже перечислены наиболее часто используемые операции фильтрования:

|  |  |
| --- | --- |
| Операции фильтрования | |
| Операция | Описание |
| <, >, <= и >= | Выполняют сравнение более чем одного значения. Эти сравнения могут быть числовыми (с данными числовых типов) или алфавитными сравнениями по словарю (со строковыми данными) |
| <> и = | Выполняют проверку на эквивалентность |
| NOT | Обращает логическое выражение. Может использоваться в сочетании с любой другой конструкцией |
| BETWEEN | Указывает включающий диапазон. Например, Units BETWEEN 5 AND 15 выбирает строки, у которых значение столбца Units находится в пределах от 5 до 15 |
| IS NULL | Проверяет столбец на null-значение |
| IN (a,b,c) | Краткая форма операции OR с одним и тем же полем. Проверяет эквивалентность значения столбца любому из перечисленных значений (a,b,c) |
| LIKE | Выполняет проверку соответствия строкового значения шаблону |
| + | Складывает два числа или выполняет конкатенацию двух строк |
| - | Вычитает одно числовое значение из другого |
| \* | Перемножает два числовых значения |
| / | Делит одно числовое значение на другое |
| % | Вычисляет модуль (остаток от деления одного числа на другое) |
| AND | Комбинирует более одной логической конструкции. Для отображения запись должна соответствовать всем критериям, объединенным с помощью AND |
| OR | Комбинирует более одной логической конструкции. Для отображения запись должна соответствовать хотя бы одному критерию из объединенных с помощью OR |

Следующий пример страницы включает три элемента управления GridView. Каждый из них привязан к одной и той же DataTable, но с разными установками фильтрации:

***private void btRead\_Click(object sender, EventArgs e)***

***{***

***// Строка подключения***

***string connectionString = "Data Source=MyPC;Initial Catalog=trainingDB;Integrated Security=True";***

***SqlConnection con = new SqlConnection(connectionString);***

***string cmd = "SELECT ProdID, catID, prodName FROM Products";***

***SqlDataAdapter adapter = new SqlDataAdapter(cmd, con);***

***DataSet dataset = new DataSet();***

***adapter.Fill(dataset, "Products");***

***// Фильтровать товар категории 2***

***DataView view1 = new DataView(dataset.Tables["Products"]);***

***view1.RowFilter = "catID=2";***

***dataGridView1.DataSource = view1;***

***label1.Text = "Filter = catID = 2";***

***// Фильтровать товары категорий 1 и 2***

***DataView view2 = new DataView(dataset.Tables["Products"]);***

***view2.RowFilter = "catID=2 or catID=1";***

***dataGridView2.DataSource = view2;***

***label2.Text = "Filter = catID = 1 or catID=2";***

***// Фильтровать товары, чье название начинается с буквы w***

***DataView view3 = new DataView(dataset.Tables["Products"]);***

***view3.RowFilter = "prodName LIKE 'w%'";***

***dataGridView3.DataSource = view3;***

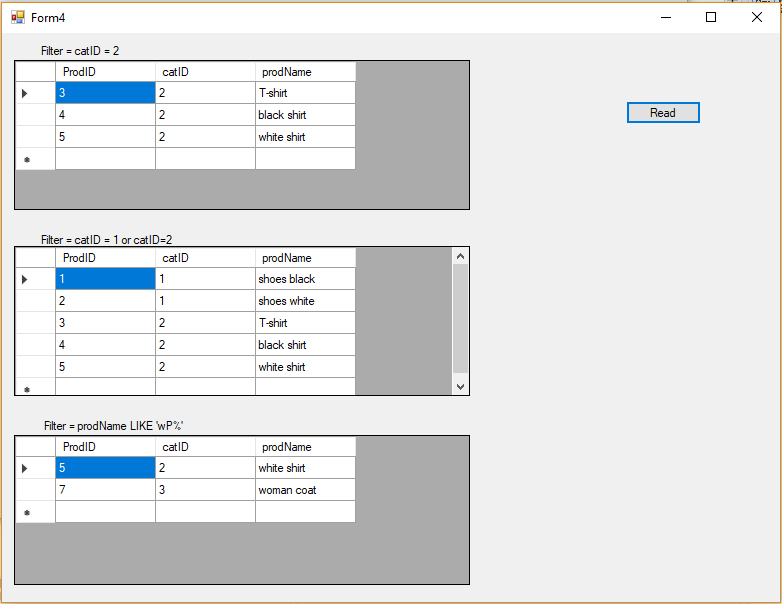
***label3.Text ="Filter = prodName LIKE 'wP%'";***

***// Скопировать данные из привязанных DataSet (DataView)***

***// Page.DataBind();***

}

Запуск этой страницы приводит к заполнению трех экранных таблиц, как показано на рисунке:



* 1. **Вычисляемые столбцы**

В дополнение к полям, извлеченным из источника данных, можно добавить вычисляемые столбцы. Вычисляемые столбцы игнорируются при извлечении и обновлении данных. Вместо этого они представляют значения, которые вычисляются на основе комбинации существующих значений. Чтобы создать вычисляемый столбец, нужно просто создать новый объект DataColumn (указав его имя и тип) и установить его свойство Expression. Затем этот объект DataColumn следует добавить в коллекцию Columns объекта DataTable с помощью метода Add().

В качестве примере рассмотрим столбец, использующий конкатенацию строк для комбинирования имени и фамилии в одном поле:

DataColumn fullName = new DataColumn( "FullName", typeof(string),

"LastName + FirstName");

dataset.Tables["Employees"].Columns.Add(fullName);

Разумеется, вы также можете выполнить запрос, создающий вычисляемые поля. Однако такой подход создает больше трудностей при последующем обновлении источника данных и прибавляет ему работы. По этой причине часто лучше создавать вычисляемые столбцы в DataSet.

Можно создавать вычисляемые столбцы, которые включают информацию из связанных строк, например, добавить в таблицу Categories столбец, показывающий количество связанных строк таблицы Products. В этом случае нужно сначала определить отношение с помощью объекта DataRelation. Также необходимо использовать агрегатную функцию SQL, такую как AVG(), MAX(), MIN() или COUNT().

В следующем примере создаются три вычисляемых столбца, каждый из которых использует агрегатную функцию и отношение таблиц:

***private void btRead\_Click(object sender, EventArgs e)***

***{***

***// Строка подключения из файла web.config***

***// Создание строки подключения***

***string connectionString = "Data Source=MYPC;Initial Catalog=trainingDB;Integrated Security=True";***

***SqlConnection con = new SqlConnection(connectionString);***

***string sqlCat = "SELECT catid, catName FROM Categories";***

***string sqlProd = "SELECT prodName, catID, price FROM Products";***

***SqlDataAdapter adapter = new SqlDataAdapter(sqlCat, con);***

***DataSet dataset = new DataSet();***

***adapter.Fill(dataset, "Categories");***

***adapter.SelectCommand.CommandText = sqlProd;***

***adapter.Fill(dataset, "Products");***

***// Связатьтаблицы Categories и Products в DataSet***

***DataRelation relation = new DataRelation("CatProds",***

***dataset.Tables["Categories"].Columns["catID"],***

***dataset.Tables["Products"].Columns["catID"]);***

***dataset.Relations.Add(relation);***

***// Создатьвычисляемыестолбцы***

***DataColumn count = new DataColumn(***

***"Кол-во продуктов", typeof(int), "COUNT(Child(CatProds).catID)");***

***DataColumn max = new DataColumn(***

***"Самый дорогой продукт: стоимость", typeof(decimal), "MAX(Child(CatProds).price)");***

***DataColumn min = new DataColumn(***

***"Самый дешевый продукт: стоимость", typeof(decimal), "MIN(Child(CatProds).price)");***

***// Добавитьстолбцы***

***dataset.Tables["Categories"].Columns.Add(count);***

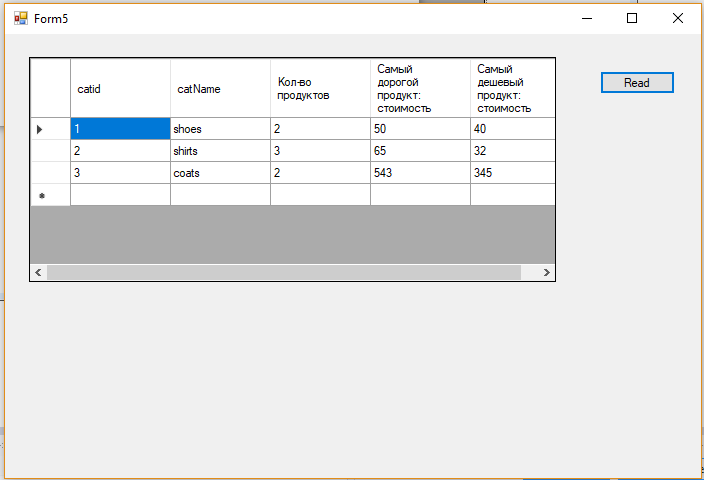
***dataset.Tables["Categories"].Columns.Add(max);***

***dataset.Tables["Categories"].Columns.Add(min);***

***// Отобразитьданные***

***dataGridView1.DataSource = dataset.Tables["Categories"];***

}На рисунке ниже показана результирующая страница:



Имейте в виду, что эти примеры просто демонстрируют удобные способы фильтрации и агрегирования данных. Эти операции являются только частью правильного представления данных. Другая часть — это их корректное форматирование.

**Задание на лабораторную работу**

1. Разработать форму (WindowsForms) для отображения данных одной из таблиц базы данных (из индивидуального задания).

2. Разместить в таблице два представления одной и той же таблицы, отсортированные по разным полям.

3. Разместить в таблице два списка для выбора параметров фильтрации записей и организовать следующую логику фильтрации:

- если в списках фильтрации ничего не выбрано, то выводятся все записи;

- если выбран параметр фильтрации в одном из списков, то фильтрация производится по этому параметру;

- если выбраны параметры фильтрации в обоих списках, то используются два условия фильтрации, связанные функцией И;

4. Создать в одной из таблиц набора DataSet вычисляемый столбец и вывести его значения в форму.

**Содержание отчета**

1. Тема и цель работы
2. Задание на лабораторную работу
3. Содержимое таблицы базы данных.
4. Внешний вид формы и программный код для каждого задания.
5. Выводы

**Контрольные вопросы**

1. Для чего предназначен объект DataView?
2. Как организуется сортировка данных с помощью DataView?
3. Что такое фильтрация данных?
4. Как организуется фильтрация данных с помощью DataView?
5. Как можно добавить строку в объект DataView?