

Лабораторная работа № 8

Тема: Разработка приложения с фильтрацией данных с помощью хранимых процедур.

Программное обеспечение: MS Visual Studio 2010

Цель: научиться создавать приложение, позволяющее с помощью хранимых процедур выполнять фильтрацию данных.

Время на выполнение: 2 академических часа.

Задание 1: Создайте хранимые процедуры:

- Student_Select_Faculty – для выборки данных из таблицы Student с параметром для фильтрации по специальности;
- Course_Select – для получения списка дисциплин;
- Grade_Select_Course – для выборки данных из таблицы Grade с параметром для фильтрации по дисциплине.

Порядок выполнения:

1. Запустите среду SQL Server Management Studio, войдите в узел своей базы данных.
2. Создайте запрос и напишите код для создания хранимой процедуры Faculty_Select.

```
create proc Student_Select_faculty
@FacultyId int
as
select * from Student
where FacultyId=@FacultyId
```

3. Код создания хранимой процедуры Grade_Select_Course:

```
create proc Grade_Select_Course
@CourseId int =1
as
select Student.FirstName, Student.LastName, Course.CourseTitle, GradeValue, Date
from (Grade inner join Student on Grade.StudentId=Student.StudentId )
inner join Course on Grade.CourseId=Course.CourseId
where Grade.CourseId=@CourseId
```

Задание 2: Разработать приложение для фильтрации данных в таблице Students по специальности.

Порядок выполнения:

1. Откройте проект, созданный в предыдущей лабораторной работе.

2. Измените дизайн приложения согласно таблице 9.1, как показано на рисунке 8.1.

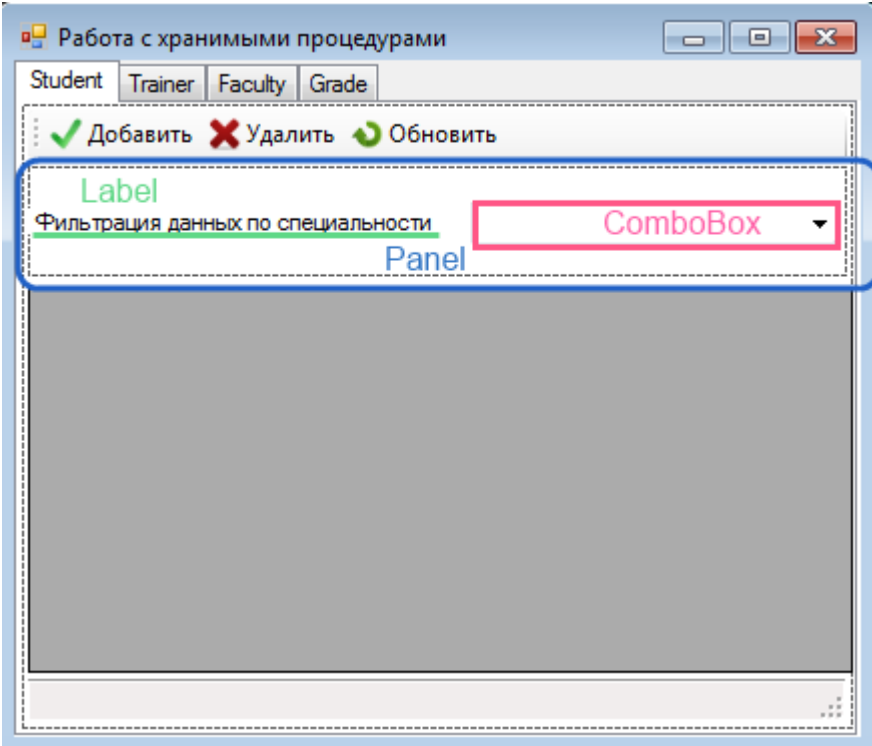


Рисунок 8.1 — Дизайн формы приложения

Таблица 9.1 — Дополнительные к приложению элементы управления формы

Элемент управления	Свойство	Значение
Panel	-	-
Label	Text	Фильтрация данных по специальности
comboBox	-	-

3. Для заполнения списка значениями необходимо вывести результат хранимой процедуры `Select_Faculty`, созданной ранее в 6 лабораторной работе. Для этого в обработчик события `Load` формы необходимо добавить код:

```

SqlConnection conn_f = new SqlConnection(Properties.Settings.Default.ConString);
SqlCommand cmd_f = new SqlCommand();
cmd_f.Connection = conn_f;
cmd_f.CommandType = CommandType.StoredProcedure;
cmd_f.CommandText = "Faculty_Select";
conn_f.Open();
SqlDataReader rdrf = cmd_f.ExecuteReader();
DataTable dtf = new DataTable();
for (int i = 0; i < rdrf.FieldCount; i++)
{
    dtf.Columns.Add(new DataColumn(rdrf.GetName(i), rdrf.GetFieldType(i)));
}
while (rdrf.Read())
{
    DataRow r = dtf.NewRow();
    for (int i = 0; i < rdrf.FieldCount; i++)
    {
        r[i] = rdrf.GetValue(i);
    }
    dtf.Rows.Add(r);
}
comboBox1.DataSource = dtf;
comboBox1.ValueMember = "FacultyId";
comboBox1.DisplayMember = "FacultyTitle";
comboBox1.SelectedValue = -1;
conn_f.Close();

```

4. Для фильтрации данных по выбранной специальности необходимо вывести результат хранимой процедуры Student_Select_Faculty, параметр хранимой процедуры связать с выбранным значением в списке comboBox1. Для этого необходимо создать обработчик события элемента comboBox1, которое возникает при выборе элемента из раскрывающегося списка – SelectionChangeCommitted (). Код обработчика события SelectionChangeCommitted представлен ниже.

```

private void comboBox1_SelectionChangeCommitted(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(Properties.Settings.Default.ConString);
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "Student_Select_faculty";
    cmd.Parameters.Add(new SqlParameter("@FacultyId", SqlDbType.Int));
    cmd.Parameters["@FacultyId"].Value = comboBox1.SelectedValue;
    conn.Open();
    |
    SqlDataReader rdr = cmd.ExecuteReader();
    DataTable dt = new DataTable();
    for (int i = 0; i < rdr.FieldCount; i++)
    {
        dt.Columns.Add(new DataColumn(rdr.GetName(i), rdr.GetFieldType(i)));
    }
    while (rdr.Read())
    {
        DataRow row = dt.NewRow();
        for (int i = 0; i < rdr.FieldCount; i++)
        {
            row[i] = rdr.GetValue(i);
        }
        dt.Rows.Add(row);
    }
    conn.Close();
    dataGridView1.DataSource = dt;
    lbRecordCount.Text = "Количество записей: " + String.Format("{0} записей", dt.Rows.Count);
}

```

5. Запустите приложение и выполните фильтрацию данных (см. рисунки 8.2 — 8.3).

	StudentId	FirstName	LastName	FacultyId
▶	1001	Савульчик	Алина	1
	1002	Сапаров	Дмитрий	1
	1003	Николаев	Никита	2
*				

Рисунок 8.2 — Главная форма в режиме отладки

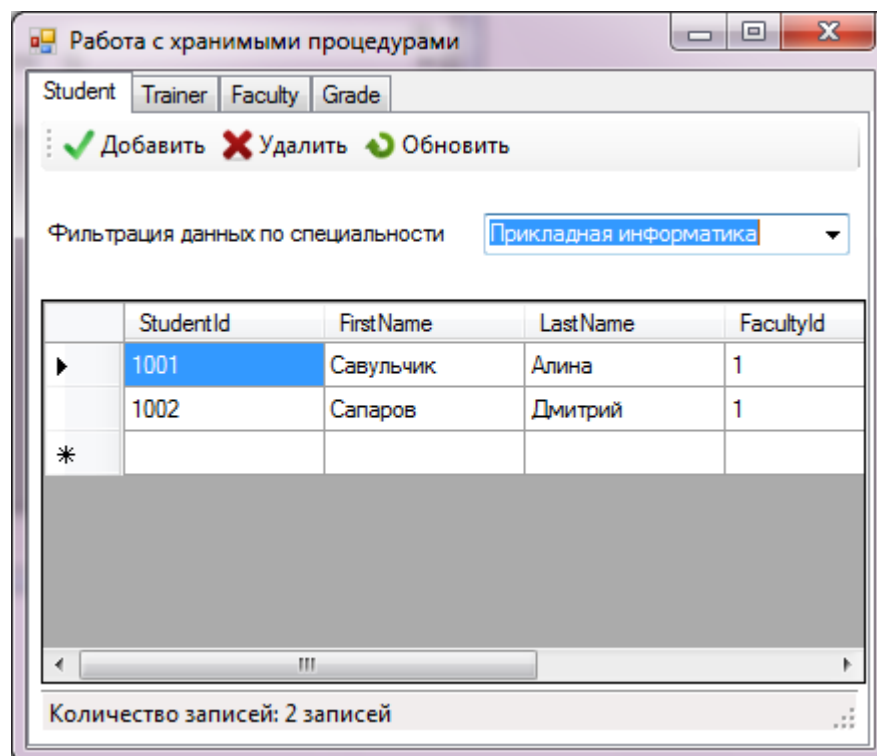


Рисунок 8.3 — Проверка фильтрации данных

Задание 3: Разработать приложение для фильтрации данных в таблице Grade по дисциплине.

Результат выполненного задания представлен на рисунке 8.4.

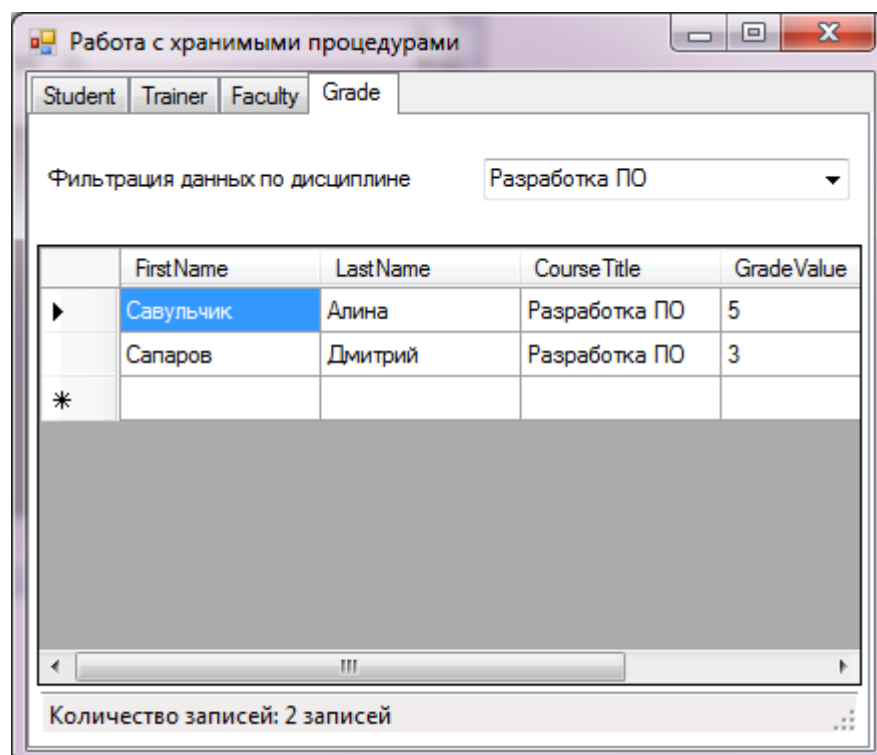


Рисунок 8.4 — Проверка фильтрации данных для таблицы Grade