

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Казанский национальный исследовательский технологический университет»**

**(ФГБОУ ВО «КНИТУ»)**

Кафедра\_Интеллектуальных систем и управления информационными ресурсами\_

Направление\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_02.03.03\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_432121\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Уровень образования\_\_\_\_\_\_\_\_\_\_\_\_бакалавр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(бакалавр, специалист, магистр)

Вид ВКР\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( проектный, исследовательский, комбинированный)

Тема Разработка веб-ориентированной информационной системы для\_\_\_\_

оценки успеваемости учащихся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Зав. кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.П. Кирпичников)

Нормоконтролер\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.Р. Мангушева)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.П. Кирпичников)

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А. И. Калислямова)

Консультанты:

по библиографии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Т.В. Толок)

2016 г.

Содержание:

Введение

1. Исследовательская часть
   1. Описание предметной области
   2. Особенности создания страниц на технологии asp.net
   3. Особенности создания и обращение к базе данных
2. Конструкторская часть
3. Технологическая часть
4. Тестирование

Заключение

Список используемой литературы и интернет-источники

Приложения

Введение

Целью данной выпускной квалификационной научно-исследовательской работы является создание веб-ориентированной информационной системы для оценки успеваемости учащихся. Система позволяет просматривать и вносить изменения в базу данных учащихся и анализировать их успеваимость.

Данная работа разработана на технологии ASP.NET, ASP.NET – это часть технологии.NET, используемая для написания мощных клиент – серверных интернет приложений. Она позволяет создавать динамические страницы HTML. Динамическими принято называть web – страницы, которые перед отправкой клиенту проходят цикл обработки на сервере. ASP.NET возникла в результате объединения более старой технологии ASP и .NET Framework. Данная работа состоит из пяти динамических страниц, которые взаимодействуют с базой данных. Для работы с базой данных используется язык структурированных запросов – SQL (Structured Query Language). Для удобного редактирования базы данных иcпользовалась Microsoft SQL 2014 Managament Studio, с помощью конструирования диаграмм баз данных и таблиц, с использованием визуальных инструментов для баз данных. Хранится база данных в Microsoft SQL 2013 Server Management Studio.

Существование проблемы оценки успеваемости учащихся в учебном заведении, позволяет заключить, что исследование, объектом которого является веб-ориентированнная информационная система для оценки успеваемости учащихся на предмет исследования учебное заведение актуально.

Практическая значимость курсового проекта заключается в том, что результаты исследования, при усовершенствовании веб-ориентированной информационной системы для оценки успеваемости учащихся, могут быть использованы в учебных заведениях для оценивания учащихся. Например, в университете, проставление оценок за итог семестра в онлановых журналах, улучшит взаимосвязь кафедры и деканата, ускорит процесс простовления оценок и закрытие сессии студентам.

Далее будет описана иследовательская, конструкторская и технологическая части, а также тестирование системы.

В иследовательской части будет описана предметная область, особенности создания страниц на ASP.NET и создание и обрание к базе данных. Будет проведенно подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Данное описание позволит корректно определить все взаимосвязи между объектами предметной области. В общем случае существуют два подхода к выбору состава и структуры предметной области:

* функциональный подход — он реализует принцип движения «от задач» и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая база данных. В этом случае мы можем четко выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны;
* предметный подход — когда информационные потребности будущих пользователей базы данных жестко не фиксируются. Они могут быть многоаспектными и весьма динамичными. Мы не можем точно выделить минимальный набор объектов предметной области, которые необходимо описывать. В описание предметной области в этом случае включаются такие объекты и взаимосвязи, которые наиболее характерны и наиболее существенны для нее. База данных, конструируемая при этом, называется предметной, то есть она может быть использована при решении множества разнообразных, заранее не определенных задач.

Также будут описаны особенности используемой технологии и использование и создание базы данных.

В конструкторской части будет описана структура базы данных и принципы работы каждой страницы. Структурирование - это набор соглашений о способах представления данных. Структурировать информацию можно по-разному. В зависимости от структуры различают иерархическую, сетевую, реляционную, объектно-ориентированную и гибридную модели баз данных. В данной работе используется реляционная структура базы данных.

В технологической части будет продемонстрировано умение использовать современные информационные технологии и информационные системы для решения практических задач в производстве и при управлении.

Кафедра \_\_\_\_\_\_\_\_ИСУИР\_\_\_\_\_\_ «УТВЕРЖДАЮ»

Направление \_\_\_\_02.03.03\_\_\_\_\_\_ Зав. кафедрой\_\_\_\_\_\_\_\_\_\_ Группа \_\_\_\_\_\_\_\_\_\_432121\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

**З А Д А Н И Е**

на выпускную квалификационную работу студента \_Калислямовой А. И. \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема:Разработка веб-ориентированной информационной системы для оценки успеваемости учащихся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Срок представления работы к защите "\_\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Цель, задачи и исходные данные работы: **\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание по разделам работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание графическойчасти (иллюстративного материала): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Консультанты:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Дата выдачи задания "\_\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.П. Кирпичников)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (А.И. Калислямова)

**ЛИСТ НОРМОКОНТРОЛЯ**

1. Лист является обязательным приложением к пояснительной записке дипломного (курсового) проекта.
2. Нормоконтролер имеет право возвращать документацию без рассмотрения в случаях:
   * нарушения установленной комплектности,
   * отсутствие обязательных подписей,
   * нечеткого выполнения текстового и графического материала.
3. Устранение ошибок, указанных нормоконтролером, обязательно.

П е р е ч е н ь

замечаний и предложений нормоконтролера по дипломному (курсовому) проекту студента

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_432121, А. И. Калислямова \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Группа, инициалы, фамилия)

|  |  |  |
| --- | --- | --- |
| Лист (страница) | Условное обозначение  (код ошибок) | Содержание замечаний и предложений со ссылкой на нормативный документ, стандарт или типовую  документацию |
|  |  |  |

Дата «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Р. Мангушева

1. Исследовательская часть

* 1. Описание предметной области

В целях ускорения и повышения эффективности процесса оценивать учащихся и следить за их успеваимостью, ставится задача разработки веб-ориентированной информационной системы для оценки успеваемости учащихся.

Веб-ориентированная информационная система должна позволять просматривать успеваемость учащихся, добавлять новые предметы, добавлять новых учеников и проставлять оценки учащимся.

База данных содержит информацию о учениках, предметах и оценках учащихся.

Данная работа использует технологию APS.NET, который позволяет выполнять обработку событий, вследствие чего старицы являются динамическими. При обработке некоторых событий, происходит соединение с базой данных и выполнение различных запросов и операций. Следствием таким запросов может быть изменения базы данных, добавление новых данных в базу, вывод данных из базы на дисплей. Для взаимодействия с базой данных используется пространство имен System.Data.SqlClient.

При написании выпукной квалификационной рабете были применены такие методы научного исследования, как изучение научной литературы по теме исследования, нормативно-правовой базы, аналитический и сравнительный методы.

1.2 Особенности создания страниц на языке APS.NET

Microsoft .NET Framework – это платформа для создания, развертывания и запуска Web-сервисов и приложений. Она предоставляет высокопроизводительную, основанную на стандартах, многоязыковую среду, которая позволяет интегрировать существующие приложения с приложениями и сервисами следующего поколения, а также решать задачи развертывания и использования интернет-приложений. .NET Framework состоит из трех основных частей - общеязыковой среды выполнения (common language runtime), иерархического множества унифицированных библиотек классов и компонентной версии ASP, называемую ASP.NET.

ASP.NET – это часть технологии .NET, используемая для написания мощных клиент-серверных интернет приложений. Она позволяет создавать динамические страницы HTML. ASP.NET возникла в результате объединения более старой технологии ASP (активные серверные страницы) и .NET Framework. Она содержит множество готовых элементов управления, используя которые можно быстро создавать интерактивные web-сайты. Вы также можете использовать сервисы, предоставляемые другими сайтами, прозрачно для пользователей вашего сайта. В общем, возможности ASP.NET ограничены только вашим воображением.

Динамическими принято называть web-страницы, которые перед отправкой клиенту проходят цикл обработки на сервере. Динамические страницы могут подстраиваться под конкретного пользователя, а также реагировать на его действия в браузере.

Любая страница состоит из двух частей. Первая часть – Design, представляет собой пустое пространство, на котором можно разместить элементы управления. Вторая часть – Source, представляет собой код языка ASP.NET. Добавление нового кода, происходит при добавлении нового объекта в окне Design, либо при ручном заполнении. Для того чтобы добавить новый элемент в окно Design, его необходимо выбрать из вкладки Toolbox, и перетащить на форму. По аналогии с C#, имеется окно Properties, в котором можно задать те или иные параметры для элементов управления, или задать те или иные события.

Каждая страница начинается со следующего кода: « <% Page Language=”текущий язык программирования” %> ». Тег “<%” всегда предназначен для интерпретации ASP – кода. Атрибут Language – это указание, что в скриптах данной страницы будет использоваться определенный язык программирования. В данной выпукной квалификационной работе – язык программирования C#. Затем идет код вида: « <form runat=”server”> ». Этот тег дает указание компилятору обрабатывать элементы управления страницы. Данный тег имеет свойство runat, для которого установлено значение “server” (других значений не бывает). При использовании этого свойства элемент управления обрабатывается компилятором, а не передается браузеру «как есть».

При двойном щелчке на объекте в окне Design откроется файл, с названием аналогичным названию страницы, но отличающийся от нее тем, что добавляется выражение типа «.cs» в конце расширения файла. Также появится код, который соответствует событию по умолчанию для данного элемента. В данном файле ведется разработка кода уже не на языке ASP.NET, а на том языке, который вы выбрали, для данной работы – это язык программирования C#. Задание и обращение к свойствам, методам ничем не отличается от обращения к ним же при написании любой другой программы на конкретно выбранном языке.

Используемый метод Response.Write() отправляет в браузер HTML код. Этот метод является полезным при выведении коротких сообщений. К примеру, он может использоваться при проверке введенных значений для вывода результата.

Карта сайта

Контент плейс

Три вью

* 1. Особенности создания и обращения к базам данных

Базы данных прошли долгий путь развития. Вначале все данные хранились в простых файлах. По мере увеличения объемов данных встал вопрос о том, как получить быстрый доступ к нужной информации. Для этого данные стали индексироваться. Другой вопрос – как избежать дублирования, когда одни и те же данные хранятся в разных местах. Сегодня мощная промышленная СУБД немыслима без систем защиты информации, журналирования, транзакции и хранимых процедур.

Данные в СУБД хранятся в таблицах. Таблица состоит из полей и записей. Запись – единица хранения данных, строка таблицы. Для работы с базами данных используется язык структурированных запросов – SQL (Structured Query Language). Команды этого языка называются запросами. Запросы служат для получения данных, для создания и изменения структуры таблиц, добавления, удаления и обновления записей и многого другого. Последовательность команд может храниться прямо на сервере СУБД в виде хранимой процедуры. Либо задаваться в виде набора команд, которые всегда проходят стадию компиляции.

Для создания базы данных в окне Sever Explorer нужно нажать на пункт Data Connections правой кнопкой мыши, и в контекстном меню выбрать Create New Sql Database.

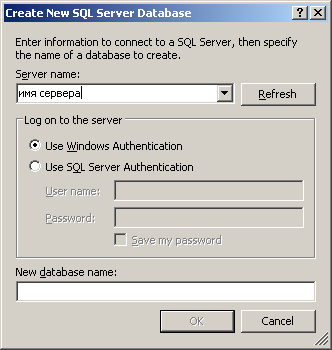


Рис. 1. Создание базы данных

На (рис. 1) показан пример создания базы данных. Для того чтобы указать компилятору, что база находится на локальном компьютере, в поле «Server name» нужно вписать «.\SQLExpress» или «(local)». Имя базы данных указывается в графе «New database name». После нажатия кнопки «ОК», база данных сохранится под указанным именем с расширением «.mdf».

После создания базы данных, в ней необходимо создать таблицы для хранения данных. Таблицу можно сделать и программно, и через окно «Server Properties». При выполнении такой последовательности действий появится узел созданной базы. Его нужно раскрыть, и щелкнуть мышью на пункт Tables, в контекстном меню нужно выбрать «Add New Table». Появится диалоговое окно, в котором нужно указать название столбца, тип данных, которые будут храниться в нем, и разрешение на хранение в ячейке нулевых значений. Бывают такие ситуации, что при создании новых строк, значения в каком-то определенном столбце автоматически увеличивались, этого можно добиться путем выбора пункта «Set Primary Key».

**Объект Connection**

Объект Connection для соединения с базой данных нуждается в строке соединения для указания пути к СУБД. Свойства класса Connection указаны в таблице:

|  |  |
| --- | --- |
| Свойство | Описание |
| DataSource | Имя экземпляра базы сервера при использовании SqlConnection |
| Database | Возвращает имя базы данных, используемой в объекте Connection после открытия |
| AttachDbFilename | Указывает путь к файлу, который является локальной базой данных |
| ConnectionString | Строка соединения с СУБД |

**Метод Open()**

Данный метод вызывается после создания объекта SqlConnection, с заданной строкой соединения - ConnectionString. Метод Open() пытается установить соединения с базой данных. Если соединения установить не удается, то появляется соответствующее окно с сообщением об ошибке.

**Использование объекта SqlCommand**

Объект SqlCommand исполняет запрос SQL, который может быть в форме встроенного текста, процедуры сервера или прямого доступа к таблице. Если это запрос на выборку данных SELECT, то данные обычно помещают в DataSet или в DataReader, но также можно производить и ручные манипуляции. Запросом на добавление данных является INSERT, на изменение UPDATE, на удаление DELETE.

Процедуры сервера нуждаются в параметрах. Они хранятся в коллекции Parametrs и имеют тип SqlParametr. Текстовые команды также могут получать параметры, перед которыми ставится префикс @.

"SELECT id FROM [sTudenT] WHERE SecondName = @DropDownList1.Text"

Или же можно использовать форматированный вывод:

string.Format("SELECT id FROM [sTudenT] WHERE SecondName = '{0}'", DropDownList1.Text);

**Метод ExecuteNonQuery()**

Данный метод применим к объектам типа SqlCommand. С помощью данного метода можно выполнять любые операции с базами данных, которые не связаны с запросом или получением данных, например, обновление, удаление записей, создание и изменение таблиц, создание процедур сервера. Она возвращает количество измененных записей в том случае, если выполняются команды Select, Update, Delete.

**Метод ExecuteScalar()**

Данный метод применим к объектам типа SqlCommand. Этот метод возвращает результат запроса в случае, если это единственное значение.

Рассмотрим пример:

string.Format("SELECT id FROM [sTudenT] WHERE SecondName = '{0}'", DropDownList1.Text);

В результате будет возвращено значение поля “id” базы даных, с параметром SecondName равным значению, выбранному из списка. Данный метод работает гораздо быстрее метода ExecuteReader().

**Метод ExecuteReader()**

Данный метод выполняется, если нужно получить табличные данные. Результат выполнения – курсор, в котором можно двигаться только от начала до конца.

В результате выполнения метода ExecuteReader() объекта Command создается объект DataReader. После использования данного метода необходимо закрывать соединения с базой, иначе такой процесс останется активным и будет занимать ресурсы. Закрыть соединение можно двумя способами. Первый – вызвать перегруженный метод ExecuteReader(), который принимает параметр типа CommandBehavior сл значением CommandBehavior.CloseConnection. В таком случае необходимо перелистать полученную выборку от начала до конца, и соединение закроется, когда будет достигнут конец.

В Visual Studio 2005 придусмотрен интерфейс автоматического вывода определнного столбца базы данных в такие элементы управления, к примеру, как DropDownList. На следующий рисунках изображен пример такого интерфеса:

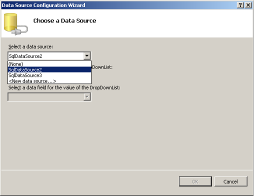


Рис. 2. Выбор источника данных

Как видно на (рис. 2), сперва, выбирается источник данных. Этот источник создается автоматически.

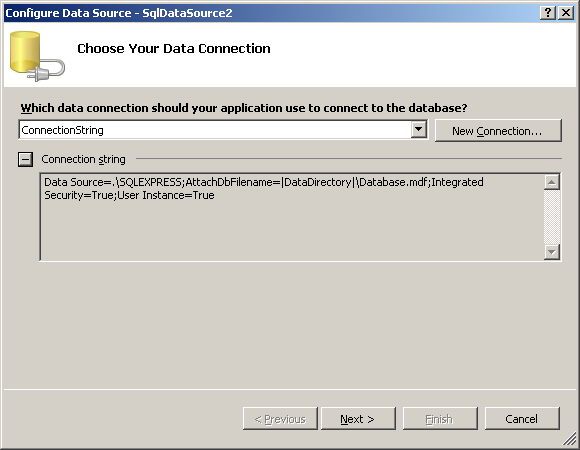


Рис. 3. Указание пути к базе данных

На (рис. 3) выбирается строка присоединения к базе данных.

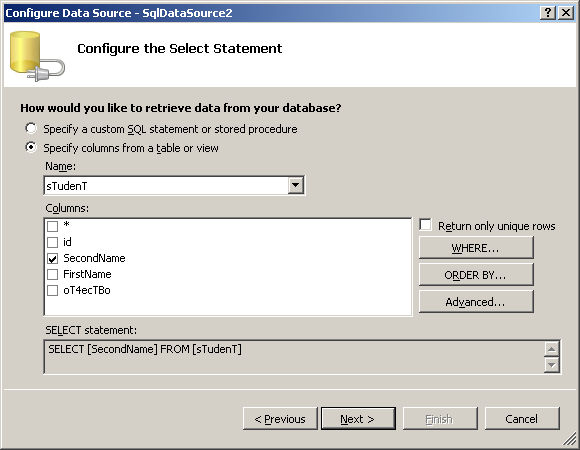


Рис. 4. Выбор столбца

На (рис. 4) выбирается столбец, который будет отображен в конкретном элементе управления. При компиляции страницы, в выбранном элементе управления, будет находится содержания выбранного столбца. Такой метод значительно упрощает разработку приложений.