Список использованной литературы:

1. Стивен Сандерсон - ASP .NET MVC Framework с примерами на C#
2. Александр Бондарь - Microsoft SQL Server 2014.
3. http://www.med.gov.kg/ - Сайт Министерства здравоохранения КР от куда брались нормативы и документы для нашей работы.
4. Жапаров М.Т., Саламатников А.С. Особенности создания базы данных для системы электронного лицензирования МЗ КР. Журнал Института геомеханики и освоения недр НАН КР «Современные проблемы механики/гидрогазодинамика, геомеханика, геотехнологии и информатика» Вып.36(2), Бишкек, 2019 г.
5. http://lisenzya.med.kg/index.php/ru/dokumenty-dlya-polucheniya-litsenzii (Документы для получения лицензии).
6. http://lisenzya.med.kg/index.php/ru/normativno-pravovye-dokumenty (Нормативно правовые документы).

References:

1. Stephen Sanderson-ASP .NET MVC Framework with examples in C#
2. Alexander Bondar-Microsoft SQL Server 2014.
3. http://www.med.gov.kg/ - Website of the Ministry of health of the Kyrgyz REPUBLIC from where the standards and documents for our work were taken.
4. Zhaparov M. T., Salamatnikov A. S. Features of creating a database for the electronic licensing system of the Ministry of health of the Kyrgyz REPUBLIC. Journal of the Institute of geomechanics and subsoil development of the national Academy of Sciences of the Kyrgyz REPUBLIC "Modern problems of mechanics/hydro-gas dynamics, geomechanics, geotechnologies and Informatics" Issue 36(2), Bishkek, 2019
5. http://lisenzya.med.kg/index.php/ru/dokumenty-dlya-polucheniya-litsenzii (Documents for obtaining a license).
6. http://lisenzya.med.kg/index.php/ru/normativno-pravovye-dokumenty (Normative legal documents).

Рецензент: к.т.н., доцент Абдулаев А.А.

УДК 681.518.3
DOI 10.33514/ВК-1694-7711-2019-2-83-92

Исмаилов Арлан Акимович, Тороев Асылбек Абакирович

Н. Исанов атындагы КМАКТАУнун маалыматтык системалар жана технологиясы кафедрасынын магистранты,

Н. Исанов атындагы КМАКТАУнун маалыматтык системалар жана технологиясы кафедрасынын доценти

Исмаилов Арлан Акимович, Тороев Асылбек Абакирович
Магистрант кафедры информационные системы и технологии КГУСТА им. Н.Исанова
доцент кафедры информационные системы и технологии КГУСТА им. Н.Исанова

Ismailov Arlan Akimovich, Toroev Asylbek Abakirovich
master's student of the Department of information systems and technologies of KSUCTA named after N. Isanov,
associate Professor of the Department of information systems and technologies of the KSUCTA named after N. Isanov

ADO. NET МААЛЫМАТТАР БАЗАСЫНДА БЕРИЛИШТЕРГЕ ЖЕТҮҮ ТЕХНОЛОГИЯЛАРЫ

ТЕХНОЛОГИИ ДОСТУПА К ДАННЫМ В ADO. NET

ADO. NET DATA ACCESS TECHNOLOGIES

Аннотация: Макалада маалыматтар базасында берилиштерге жетүү технологиялары алардын артыкчылыктарын жана кемчиликтерин эске алуу менен талкууланат. Долбоордун маалымат базасында берилиштерге жетүүнү уюштуруу үчүн ADO. NET технологиясы тандалды. ADO. NET технологиясы. NET платформасынын курамында MS VC # дизайн чөйрөсү менен, ошондой эле VC # инструменталдык чөйрөсү менен толук шайкештикти камсыз кылат, бул дизайн инструменталдык чөйрөсүнүн кең мүмкүнчүлүктөрүн колдонууга мүмкүндүк берет.

Аннотация: В статье рассматриваются технологии доступа к данным с учетом их достоинств и недостатков. Выбрана технология ADO. NET для организации доступа к данным в базе данных проекта. Технология ADO. NET как часть платформы. NET обеспечивает полную совместимость со средой проектирования MS VC#, также и с инструментальной средой VC#, что позволяет использовать широкие возможности инструментальной среды проектирования.

Annotation: The article discusses data access technologies taking into account their advantages and disadvantages. ADO. NET technology was selected for organizing access to data in the project database. ADO. NET technology as part of the .NET platform provides full compatibility with the MS VC # design environment,

as well as with the VC # tool environment, which allows you to use the wide capabilities of the design tool environment.

Негизги сөздөр: NET платформасы, MS VC # дизайн чөйрөсү, маалымат берүүчү, Data Set объектиси. Sgl Data Reader объектиси.

Ключевые слова: платформа .NET, среда проектирования MS VC#, провайдер данных, объект

Keywords: NET platform, MS VC # design environment, data provider, Data Set object. Sgl Data Reader object Data Set. объект Sgl Data Reader.

MS Visual Studio. NET предоставляет большое количество компонентов, инструментальных средств, которые позволяют быстро и эффективно создать архитектуру доступа к данным во время разработки и оснастить приложение надежным механизмом доступа к данным, затратив минимум времени на формирование кода программы проекта.

Наряду с этим все возможности объектной модели ADO. NET доступны программно, что позволяет реализовать нестандартные функции или создавать приложения, ориентированные на нужды пользователя.

Доступ к данным в ADO. NET основан на использовании двух компонентов:

1. провайдера данных (Data Provider), выполняющего функции посредника при взаимодействии программы и баз данных;
2. набора данных — объекта Data Set, в котором данные хранятся на локальном компьютере [1].

Связь с базой данных создается и поддерживается при помощи провайдера данных. Провайдерами данных .NET Framework являются компоненты, которые специально сконструированы для обработки данных и быстрого, однопроходного доступа к данным только для чтения. В действительности провайдер — это набор классов, взаимосвязанных компонентов, обеспечивающих эффективный высокопроизводительный доступ к данным.

“Провайдер данных состоит из четырех компонентов:

- Connection — обеспечивает подключение к источнику данных;
- Command — применяется для управления источником данных; позволяет исполнять команды, не возвращающие данных, например INSERT, UPDATE и DELETE, либо команды, возвращающие объект Data Reader (такие как SELECT).

Объект Command позволяет обращаться к командам базы данных для возврата данных, изменения данных, выполнения хранимых процедур и передачи или получения сведений о параметрах:

- Data Reader — предоставляет доступный только для однонаправленного чтения набор записей, подключенный к источнику данных. Data Reader обеспечивает высокопроизводительный поток данных из источника данных;
- Data Adapter — заполняет отсоединенный объект Data Set или Data Table и обновляет его содержимое.

Data Adapter предоставляет мост между объектом Data Set и источником данных [1].

Data Adapter использует объекты Command для выполнения команд SQL на источнике данных, для загрузки Data Set с данными и согласования изменений данных, выполненных в Data Set, вновь с источником данных.

Полное описание объектов, компонентов в MS Visual Studio дано в наиболее полном руководстве [2].

Объект Connection представляет соединение с базой данных.

“Visual Studio поддерживает два класса Connection:

- Sql Connection, предназначенный для подключения к SQL Server;
- Ole Db Connection, который применяется для подключения к самым разным БД.

Все данные, необходимые для открытия канала связи с базой данных, хранятся в свойстве Connection String объекта Connection, этот объект также поддерживает ряд методов, позволяющих обрабатывать данные с применением транзакций.

Объект Command также представлен двумя классами:

- Sgl Command
- Ole Db Command.

Он позволяет исполнять команды над базой данных, используя для обмена данными установленное соединение.

При помощи объектов Command можно исполнять хранимые процедуры, команды SQL, а также операторы, возвращающие целые таблицы [2].

Объект Sql Data Reader предоставляет поток с набором записей базы данных, доступный только для однонаправленного чтения. В отличие от других компонентов провайдера данных, создавать экземпляры Data Reader напрямую не разрешается, его можно получить при помощи методов Execute Reader объекта Command: метод Sql Command. Execute Reader () возвращает объект Sgl Data Reader. Поскольку в любой момент времени в памяти находится только одна строка, использование объекта Sql Data Reader почти не снижает

производительность системы, но требует монопольного доступа к открытому объекту Connection в течение времени жизни объекта Sql Data Reader.

Data Adapter — это основной класс ADO. NET, обеспечивающий доступ к отсоединенным данным. В сущности, он выполняет функции посредника во взаимодействии между БД и объектом Data Set. При вызове метода Fill () объект Data Adapter заполняет Data Table или Data Set данными, полученными из базы данных. После обработки данных, загруженных в память, можно записать модифицированные данные в базу данных, вызвав метод Update () объекта Data Adapter. При вызове метода Update () все измененные данные копируются из объекта Data Set в базу данных с исполнением соответствующей команды Insert Command, Delete Command или Update Command.

Доступ к данным в ADO. NET осуществляется в такой последовательности:

1. Объект Connection устанавливает между приложением и базой данных соединение, напрямую доступное объектам Command и Data Adapter.
2. Объект Command позволяет исполнять команды непосредственно над базой данных. Если исполненная команда возвращает несколько значений, Command открывает к ним доступ через объект Data Reader.
3. Полученные результаты можно обрабатывать напрямую, используя код приложения, либо через объект Data Set, заполнив его при помощи объекта Data Adapter.

Для обновления базы данных также применяются объекты Command или Data Adapter.

Объект Data Set — это представление в памяти компьютера данных, изолированных от источника данных. Этот объект можно также рассматривать как локальную копию фрагмента БД. Объект Data Set хранится в памяти, его содержимым разрешено манипулировать и обновлять независимо от БД, играющей роль источника данных. При необходимости объект Data Set может служить шаблоном для обновления серверной БД. “Объект Data Set содержит набор объектов Data Table (этот набор может быть и пустым, т. е. не содержать ни одного объекта Data Table). Каждый объект Data Table представляет в памяти компьютера одну таблицу. Структура объекта Data Table определяется двумя наборами:

1.Data Columns, куда входят все столбцы таблицы, которую представляет объект Data Table, и набором ограничений таблицы. Вместе эти два набора составляют схему таблицы.

2.В Data Table также входит набор Data Rows, где, собственно, и хранятся данные объекта Data Set.

3. Data Set содержит набор Extended Properties, в котором хранятся дополнительные данные объекта Data Set” [2].

Класс Data Set в ADO.NET специально сконструирован для доступа к данным независимо от источника данных. Поэтому он может быть использован со многими разными источниками данных, с XML-данными или для управления данными, локальными для приложения. Data Set содержит коллекцию одного или нескольких объектов Data Table, состоящих из строк и столбцов данных, а также первичный ключ, внешний ключ, ограничение и связанные сведения о данных в объектах Data Table.

Для управления текущими соединениями с источниками данных служит окно Server Explorer. Текущие соединения с источниками данных, доступные в Visual Studio, отображаются в окне Server Explorer в виде узлов дерева Data Connections. Чтобы добавить к проекту соединение с базой данных, достаточно перетащить нужный узел Data Connections из окна Server Explorer в окно дизайнера формы.

Окно Server Explorer также позволяет создать новое соединение, щелкнув правой кнопкой узел Data Connections и выбрав из контекстного меню команду Add Connection. Окно Add Connection позволяет выбрать провайдер данных. При нажатии кнопки Change появится диалоговое окно Change Data Source, которое предоставляет вам выбрать провайдер для подключения к базе данных. проверить соединение можно, нажав кнопку Test Connection.

«Объект Connection» Создать объект Connection в коде программы:

А) создать новый объект Command и присвоить его свойству Connection String содержимое строки соединения с базой данных.

Б) Создание объекта Command в коде программы

Sql Connection con = new Sql Connection; con. Connection String = "Data Source=(local); Initial Catalog=College; Integrated Security=True";

Можно также задать строку подключения в качестве параметра в конструкторе объекта Connection: Sql Connection con = new Sql Connection (string connectionString); [2]”

Объект Command содержит ссылку на хранимую процедуру БД или оператор SQL и способен исполнить этот оператор на источнике данных, используя активное соединение. Объект Command также содержит все данные, необходимые для исполнения команды: ссылку на активное соединение, текст команды и ее параметры. Подобно другим классам компонентов провайдера данных, класс Command представлен двумя вариантами. Первый, Ole Db Command, разработан для взаимодействия с самыми разными типами БД, а второй — Sql Command — только для баз данных SQL Server. Для класса Command достаточно активного соединения, взаимодействие с объектом Data

Adapter ему не требуется. В силу этих причин объекты Command очень быстро и эффективно взаимодействуют с базами данных, позволяя выполнять различные операции:

- исполнять команды, не возвращающие значения, например INSERT, UPDATE и DELETE;
- исполнять команды, возвращающие единственное значение;
- исполнять команды языка Database Definition Language, например CREATE TABLE и ALTER;
- работать с объектом Data Adapter, возвращающим объект Data Set;
- возвращать результирующий набор непосредственно через экземпляр объекта Data Reader — это самый быстрый способ доступа к данным, он особенно удобен, если данные требуются только для чтения;
- возвращать результирующий набор в виде потока XML — эту возможность поддерживает только класс Sql Command;
- возвращать результирующий набор, созданный на основе нескольких таблиц или в результате исполнения нескольких операторов.

У объекта Command есть свойство Command Type, которое определяет тип команды, содержащийся в свойстве Command Text, оно может принимать одно из следующих значений:

- Text — заставляет рассматривать значение свойства Command Text как текст команды SQL, при этом в свойстве Command Text должен быть один или несколько допустимых операторов SQL, разделенных точкой с запятой. В последнем случае операторы SQL выполняются по порядку;
- Stored Procedure — в этом случае в свойстве Command Text необходимо указать имя существующей хранимой процедуры — она будет исполнена при вызове данной команды;
- Table Direct — в этом случае в свойстве Command Text должно быть имя одной или нескольких таблиц. При исполнении эта команда вернет все столбцы и строки таблиц, заданных свойством Command Text.
- Создание объекта Command

Sql Command cmd = new Sql Command; cmd. Connection = con; cmd. Command Type = Command Type.Text; cmd. Command Text = "SELECT * FROM Student";

Свойство Connection устанавливается в соответствии с типом соединения: для объекта Sql Command требуется соединение на основе объекта Sql Connection, а для Ole Db Command — соединение на основе Ole Db Connection. Объект Command поддерживает три метода:

- Execute Non Query — исполняет команды, не возвращающие данные, например INSERT, UPDATE и DELETE;

- Execute Scalar — исполняет запросы к БД, возвращающие единственное значение;
- Execute Reader — возвращает результирующий набор через объект Data Reader. Все эти методы исполняют на источнике данных команду, представленную объектом Command, отличаются они возвращаемым значением.

Метод Execute Non Query — самый простой из них, он не возвращает никаких значений. Этот метод обычно применяют для вызова хранимых процедур или команд SQL, таких как INSERT, UPDATE или DELETE.

Метод Execute Scalar возвращает только значение первого поля первой строки, извлеченной заданной командой, независимо от того, сколько строк выбрано этой командой в действительности.

Метод Execute Reader возвращает неизменяемый объект Data Reader, допускающий только последовательный однонаправленный просмотр без использования объекта Data Adapter. Если не требуется модифицировать содержимое базы данных или как-то иначе манипулировать им, этот способ извлечения данных является самым быстрым и эффективным.

Класс Sql Command поддерживает еще один интересный метод — Execute Xml Reader, возвращающий результирующий набор в формате XML; результаты возвращаются в виде неизменяемого объекта Xml Reader, доступного только для последовательно однонаправленного просмотра. Объект Data Reader

Исполнить с помощью объекта Command команду, возвращающую набор, можно методом Execute Reader. Он возвращает объект Data Reader. Объект Data Reader — это упрощенный объект, обеспечивающий быстрое и эффективное последовательное однонаправленное чтение данных. Объект Data Reader позволяет перебирать записи результирующего набора, передавая нужные значения напрямую коду приложения. При этом данные разрешается просматривать только в одном направлении: нельзя вернуться к записи, прочитанной ранее. Кроме того, объект Data Reader ориентирован на использование постоянных соединений и, пока он существует, требует монопольного доступа к активному соединению.

Объект Data Reader нельзя создать напрямую, это делается путем вызова метода Execute Reader объекта Command. Подобно другим членам классов провайдеров данных, у каждого класса Data Provider есть собственный класс Data Reader. Например, объект Sql Command возвращает Sql Data Reader. При вызове метода Execute Reader объект Command исполняет представленную им команду и создает объект Data Reader соответствующего типа, который можно

записать в переменную ссылочного типа. Получив ссылку на объект Data Reader, можно просматривать записи, загружая нужные данные в память. У нового объекта Data Reader указатель чтения устанавливается на первую запись результирующего набора. Чтобы сделать ее доступной, следует вызвать метод Read. Если запись доступна, метод Read переводит указатель объекта Data Reader к следующей записи и возвращает true, в противном случае метод Read возвращает false. Таким образом, метод Read используют для перебора записей в цикле while, например, так: Sql Data Reader rdr = cmd. Execute Reader; while (rdr. Read ()) {...}

Для перебора записей также можно использовать цикл foreach. Класс Sql Data Reader содержит метод Get Enumerator, и при каждом шаге цикла он будет возвращать объект Db Data Record, представляющий одну запись, который можно сохранить, например, в Array List.

Листинг 7.3. Создание Data Reader и заполнение данными объекта Array List

// открываем соединение con. Open; Sql Data Reader rdr = cmd. Execute Reader; Array List records = new Array List; if (rdr. Has Rows) {foreach (Db Data Record rec in rdr) {records.Add (rec);}} // закрываем соединение con.Close();

Выполнение метода Execute Reader требует открытого соединения с базой данных. Закончив чтение данных, вызовите метод Close объекта Data Reader, чтобы закрыть соединение.

При чтении записи с помощью объекта Data Reader значения отдельных полей доступны через индексатор или свойство по умолчанию в виде массива объектов, к элементам которого разрешается обращаться по индексу: object o = rdr[3]; или по имени поля: object o = rdr["Customer ID"]; При этом способе доступа Data Reader предоставляет все значения в виде объектов, хотя из Data Reader допустимо извлекать и типизированные данные. Более подробно об этом рассказано далее. Прочитав данные с помощью Data Reader, следует вызвать метод Close, чтобы закрыть Data Reader, в противном случае объект Data Reader будет удерживать монопольный доступ к активному соединению, сделав его недоступным другим объектам. При вызове метода Execute Reader, установив свойство Command Behavior в Close Connection, вы автоматически закроете соединение, не вызывая метод Close явно.

Список использованной литературы:

1. Фленов М.Е. Ф71 Библия C#. -3-е изд., перераб. и доп. -Спб.: Бхв-Петербург, 2016. 544 с.
2. www.msdn.com
3. Холл М. Сервлеты и Java Server Pages. Библиотека программиста. – Спб.: Питер, 2011. – 496 с.

References:

1. Flenov M. E. F71 The Bible With#. - 3rd ed., reprint. and add. - St. Petersburg: BHV-Petersburg, 2016. 544 p.
2. www.msdn.com
3. Hall M. Servlets and Java Server Pages. Programmer's library. Saint Petersburg: Piter, 2011, 496 p.

Рецензент: к.т.н., доцент Абыкеев К.Ж.

УДК 681.518.3
DOI 10.33514/ВК-1694-7711-2019-2-92-100

Исмаилов Арлан Акимович, Тороев Асылбек Абакирович
Н. Исанов атындагы КМАКТАУнун маалыматтык системалар жана технологиясы кафедрасынын магистранты,
Н. Исанов атындагы КМАКТАУнун маалыматтык системалар жана технологиясы кафедрасынын доценти

Исмаилов Арлан Акимович, Тороев Асылбек Абакирович
Магистрант кафедры информационные системы и технологии КГУСТА им. Н.Исанова
доцент кафедры информационные системы и технологии КГУСТА им. Н.Исанова

Ismailov Arlan Akimovich, Toroev Asylbek Abakirovich
master's student of the Department of information systems and technologies of KSUCTA named after N. Isanov,
associate Professor of the Department of information systems and technologies of the KSUCTA named after N. Isanov

MS VC # КОМПОНЕНТИН ИШТЕП ЧЫГУУНУ КОЛДОНУУ МЕНЕН ЭКОНОМИКА СУБЪЕКТТЕРИНИН МААЛЫМАТ ТУТУМДАРЫН ӨНҮКТҮРҮҮ

РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ СУБЪЕКТА ЭКОНОМИКИ С ИСПОЛЬЗОВАНИЕМ КОМПОНЕНТНОЙ РАЗРАБОТКИ В MS VC#