Работа с базами данных. Технология ADO.NET.

Технология ADO

- интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft (MS Access, MS SQL Server) и основанный на технологии компонентов ActiveX. ADO позволяет представлять данные из разнообразных источников (реляционных баз данных, текстовых файлов и т. д.) в объектно-ориентированном виде.

Проблемы ADO

- Наиболее заметная из них — громоздкость (в плане физического размера) отключенного набора записей. Потребность в этом средстве возрастает по мере развития веб-ориентированных вычислений, поэтому в данном случае понадобился свежий подход.

ADO.NET

- - технология, предоставляющая доступ к данным для приложений, основанных на Microsoft .NET. Является не развитием более ранней технологии ADO, а самостоятельной технологией, частью фреймворка .NET.

Три стороны ADO.NET

Библиотеки ADO.NET можно применять тремя концептуально различными способами:

- в подключенном режиме,
- в автономном режиме,
- с помощью технологии Entity Framework.

Подключенный режим

На данном уровне работа базами данных ведётся через объекты подключения, объекты чтения данных и поставщика данных предназначенного для нужной СУБД. Для получения данных выполняются следующие шаги.

- Создание, настройка и открытие объекта подключения.
- Создание и настройка объекта команды, указывающего объект подключения в аргументе конструктора или через свойство Connection.
- Вызов метода ExecuteReader () настроенного объекта команды.
- Обработка каждой записи с помощью метода Read () объекта чтения данных.
- Объекты чтения данных предоставляют поток данных, для чтения в прямом направлении. Чтение происходит каждый раз по одной записи. Следовательно объекты чтения обрабатывают только select запросы. Открытие и закрытие подключения к БД полностью возлагается на программиста.

Автономный уровень

- Автономный уровень ADO.NET позволяет отображать реляционные данные с помощью модели объектов в память. Типы данных из System.Data воспроизводят не только отображение строк и столбцов, а также отношения между таблицами, первичные ключи и т. д. Так как отображение данных происходит в память, подключение не занимает времени СУБД, подключаясь и отключаясь автоматически, при чтении и обновлении данных, автономный уровень снимает с программиста лишнюю работу. Но у данного уровня есть недостаток, представьте что требуется считать из БД 20000 записей, и при использовании автономного уровня, все это ляжет в память приложения, не очень разумное использование, здесь на выручку приходит подключаемый уровень который считает все последовательно.

Entity Framework

- Entity Framework выводит абстракцию на новый уровень - объектной модели. Теперь отображение происходит на бизнес-объекты приложения, что позволяет работать с данными как с обычными объектами языка. Сущности (entities) — это концептуальная модель физической базы данных, которая отображается на предметную область. Формально говоря, эта модель называется моделью сущностных данных (Entity Data Model — EDM). Модель EDM представляет собой набор классов клиентской стороны, которые отображаются на физическую базу данных. Тем не менее, нужно понимать, что сущности вовсе не обязаны напрямую отображаться на схему базы данных, как может показаться, исходя из названия. Сущностные классы можно реструктурировать для соответствия существующим потребностям, и исполняющая среда EF отобразит эти уникальные имена на корректную схему базы данных.

Архитектура ADO.NET

- В ADO.NET используется многоуровневая архитектура, которая обращается вокруг небольшого числа ключевых концепций, таких как объекты Connection, Command и DataSet. Однако архитектура ADO.NET серьезно отличается от классической архитекуры ADO.

Поставщики данных

- *Поставщик данных (data provider)* — это набор классов ADO.NET, которые позволяют получать доступ к определенной базе данных, выполнять команды SQL и извлекать данные.
- Однако независимо от используемого поставщика данных, каждый из них определяет набор классов, обеспечивающих основную функциональность.

Поставщики данных

- ADO.NET поставляется с тремя пространствами имен клиента базы данных: одно для SQL Server, другое для источников данных Open Database Connectivity (ODBC) и третье для любой базы данных, доступной через OLE DB.

Основные объекты поставщиков данных ADO.NET			
Тип объекта	Базовый класс	Соответствующие интерфейсы	Назначение
Connection	DbConnection	IDbConnection	Позволяет подключаться к хранилищу данных и отключаться от него. Кроме того, объекты подключения обеспечивают доступ к соответствующим объектам транзакций
Command	DbCommand	IDbCommand	Представляет SQL-запрос или хранимую процедуру. Кроме того, объекты команд предоставляют доступ к объекту чтения данных конкретного поставщика данных
DataReader	DbDataReader	IDataReader, IDataRecord	Предоставляет доступ к данным только для чтения в прямом направлении с помощью курсора на стороне сервера

Основные объекты поставщиков данных ADO.NET

Тип объекта	Базовый класс	Соответствующие интерфейсы	Назначение
DataAdapter	DbDataAdapter	IDataAdapter, IDbDataAdapter	Пересылает наборы данных из хранилища данных к вызывающему процессу и обратно. Адаптеры данных содержат подключение и набор из четырех внутренних объектов команд для выборки, вставки, изменения и удаления информации в хранилище данных
Parameter	DbParameter	IDataParameter, IDbDataParameter	Представляет именованный параметр в параметризованном запросе
Transaction	DbTransaction	IDbTransaction	Инкапсулирует транзакцию в базе данных

Название поставщиков

- Конкретные имена этих основных классов различаются у различных поставщиков (например, SqlConnection, OracleConnection, OdbcConnection и MySqlConnection), но все эти объекты порождены от одного и того же базового класса (в случае объектов подключения это DbConnection), который реализует идентичные интерфейсы (вроде IDbConnection)

Общий принцип действия

- Но даже несмотря на то, что разные поставщики данных .NET используют различные классы, все они некоторым образом стандартизированы.
- *Точнее говоря, каждый поставщик основан на одном и том же наборе интерфейсов и базовых классов. Так, например, объект Connection реализует интерфейс IDbConnection, который определяет такие ключевые методы, как Open() и Close(). Подобная стандартизация гарантирует, что каждый класс Connection будет работать одинаковым образом и предоставит один и тот же набор ключевых свойств и методов.*

Достоинства ADO.NET

- Поскольку каждый поставщик использует одни и те же интерфейсы и базовые классы, можно писать обобщенный код доступа к данным работая с интерфейсами.
- Поскольку каждый поставщик реализован отдельно, он может использовать соответствующую оптимизацию.
- Кроме того, специализированные поставщики могут добавлять нестандартные средства, которых не имеют другие поставщики (например, возможность SQL Sever выполнять XML-запросы).

Фундаментальные классы ADO.NET

- *System.Data*
- Содержит ключевые классы контейнеров данных, которые моделируют столбцы, отношения, таблицы, наборы данных, строки, представления и ограничения. Дополнительно содержит ключевые интерфейсы, которые реализованы объектами данных, основанными на соединениях

Фундаментальные классы ADO.NET

- *System.Data.SqlClient*
- Содержит классы, используемые для подключения к базе данных Microsoft SQL Server, в том числе SqlDbCommand, SqlDbConnection и SqlDbDataAdapter. Эти классы оптимизированы для использования интерфейса TDS к SQL Server

Фундаментальные классы ADO.NET

- *System.Data.OleDb*
- Содержит классы, используемые для подключения к поставщику OLE DB, включая OleDbCommand, OleDbConnection и OleDbDataAdapter. Эти классы поддерживают большинство поставщиков OLE DB, но не те, что требуют интерфейсов OLE DB версии 2.5

Фундаментальные классы ADO.NET

- *System.Data.OracleClient*
- Содержит классы, необходимые для подключения к базе данных Oracle (версии 8.1.7 и выше), в том числе OracleCommand, OracleConnection и OracleDataAdapter. Эти классы используют оптимизированный интерфейс *OCI (Oracle Call Interface — Интерфейс вызовов Oracle)*

Фундаментальные классы ADO.NET

- *System.Data.Odbc*
- Содержит классы, необходимые для подключения к большинству драйверов ODBC, такие как OdbcCommand, OdbcConnection, OdbcDataReader и OdbcDataAdapter. Драйверы ODBC поставляются для всех видов источников данных и конфигурируются через значок Data Sources (Источники данных) панели управления

Фундаментальные классы ADO.NET

- *System.Data.SqlTypes*
- Содержит структуры, соответствующие встроенным типам данных SQL Server. Эти классы не являются необходимыми, но предоставляют альтернативу применению стандартных типов данных .NET, требующих автоматического преобразования