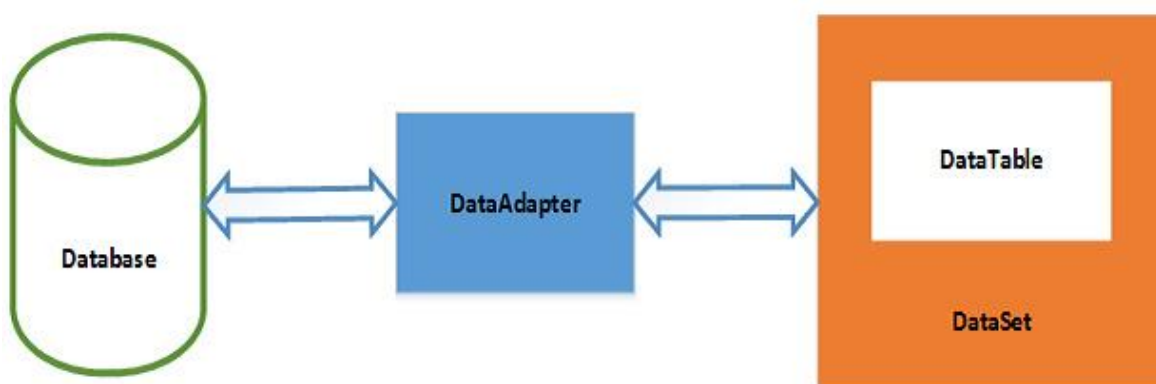


4-12-2018

Conectividad de base de datos

ADO.net



Unidad 6

Taller de base de datos

Docente: Verónica Jáuregui

Daniela itzel Hernández Muñoz 16400929
Jesús Francisco Melgarejo Saldaña 16400954
José Luis Ramos Monreal 16400984
Enrique Ortiz lora 16400967
Martin Alejandro Jiménez Díaz 16400940
Víctor Joel camberos García 16400889

La conectividad a bases de datos nos permite la comunicación entre los usuarios y los datos mediante una aplicación y el gestor de la base de datos, el software para establecer dicha conexión es individual del sistema gestor, existe un estándar (ODBC) que es un estándar de acceso a las bases de datos desarrollados en SQL su objetivo es hacer posible el acceso a cualquier dato desde cualquier aplicación sin importar que sistema gestor de se esté usando, lo hace a través de un sistema de Windows Los creadores de las distintas bases de datos son los responsables de crear un driver ODBC para que su base de datos se pueda conectar desde un sistema Microsoft.

El software funciona de dos modos, con un software manejador en el cliente, o una filosofía cliente-servidor. En el primer modo, el driver interpreta las conexiones y llamadas SQL y las traduce desde el API ODBC hacia el DBMS. En el segundo modo para conectarse a labase de datos se crea una DSN dentro del ODBC que define los parámetros, ruta y características de la conexión según los datos que solicite el creador o fabricante.

¿Qué es ado.net?

ADO.NET es un conjunto de clases que exponen los servicios de acceso a datos para los programadores de .NET Framework. ADO.NET proporciona un amplio conjunto de componentes para crear aplicaciones distribuidas de intercambio de datos.

Es una parte integral de .NET Framework, que proporciona acceso a datos relacionales, XML y de aplicaciones. ADO.NET es compatible con una variedad de necesidades de desarrollo, incluida la creación de clientes de bases de datos front-end y objetos comerciales de nivel medio utilizados por aplicaciones, herramientas, idiomas o navegadores de Internet.

Consta de dos escenarios **el conectado**

En el que se puede tener una conexión permanente con el usuario con datos recientes pero consume muchos recursos

- 1: se inicia la conexión a la BD
- 2: se ejecuta un comando
- 3: se procesan los datos
- 4 se libera el comando
- 5: se cierra la conexión

Y el desconectado

- 1) Abrir Conexión
- 2) Llenar DataSet mediante DataAdapter
- 3) Cerrar Conexión
- 4) Procesar DataSet
- 5) Abrir Conexión
- 6) Actualizar fuente de datos mediante DataAdapter
- 7) Cerrar Conexión

En el que se pueden usar datos independientes pero puede que el usuario no esté usando los datos más actuales

Ado.net consiste en 2 partes primarias

Data provider: esta clase proporcionan el acceso a una fuente de datos, como Microsoft sql server y Oracle. Cada fuente de datos tiene su propio conjunto de objetos del proveedor, pero cada uno tiene un conjunto común de clases de utilidad:

- **Connection:** Proporciona una conexión usada para comunicarse con la fuente de datos.
- **Command:** Usado para realizar alguna acción en la fuente de datos, como lectura, actualización, o borrado de datos relacionales.
- **Parameter:** Describe un simple parámetro para un *command*.
- **DataAdapter:** "Puente" utilizado para transferir data entre una fuente de datos y un objeto DataSet (ver abajo).
- **DataReader:** Es una clase usada para procesar eficientemente una lista grande de resultados, un registro a la vez.

Data sets: los objetos DataSets, son un grupo de clases que describen una simple base de datos relacional en memoria, fueron la estrella del show en el lanzamiento inicial (1.0) del Microsoft .NET Framework. Las clases forman una jerarquía de contención:

- Un objeto DataSet representa un esquema (o una base de datos entera o un subconjunto de una). Puede contener las tablas y las relaciones entre esas tablas.
- Un objeto DataTable representa una sola tabla en la base de datos. Tiene un nombre, filas, y columnas.
- Un objeto DataView "se sienta sobre" un DataTable y ordena los datos (como una cláusula "order by" de SQL) y, si se activa un filtro, filtra los registros (como una cláusula "where" del SQL).
- Un DataColumn representa una columna de la tabla, incluyendo su nombre y tipo.
- Un objeto DataRow representa una sola fila en la tabla, y permite leer y actualizar los valores en esa fila, así como la recuperación de cualquier fila que esté relacionada con ella a través de una relación de clave primaria - clave extranjera.
- Un DataRowView representa una sola fila de un DataView,
- Un DataRelation es una relación entre las tablas, tales como una relación de clave primaria - clave ajena. Esto es útil para permitir la funcionalidad del DataRow de recuperar filas relacionadas.
- Un Constraint describe una propiedad de la base de datos que se debe cumplir, como que los valores en una columna de clave primaria deben ser únicos.

Conectividad en Dispositivos Móviles

Los Sistemas Manejadores de Base de Datos Móviles(SMBDM) Han permitido que tecnologías como internet no sea únicamente accesibles a la conexión por dispositivos no móviles como un computador convencional con conexión por red alámbrica, dado que internet entro en los 90's como un boom de la World Web Wide(WWW), logrando la

interacción de un usuario móvil con una aplicación mediante Lenguajes de Etiqueta tales como XML. De esto surgen las Redes Inalámbricas en Dispositivos Móviles tales como los PDAs, Computadores Portátiles, MP3s, entre otros dispositivos permitiendo una conectividad a distancia llamándolo a esto la Computación Móvil.

Gracias a la conectividad móvil podemos recuperar información de sistemas de computación y/o repositorios de información mediante algún Dispositivo Móvil que cuente con tecnologías tales como la generación 2G en adelante en la que la Tecnología Móvil se vuelve Digital y existe la transferencia de datos mediante el uso de Internet, permitiendo a cualquier usuario manipular la información a distancia.

Esto fue considerado un boom en la década de los 90's debido a que se buscaban alternativas para que la Red mundial no fuera únicamente desde un solo sitio, se buscaba movilidad, donde un usuario en cualquier parte del mundo pudiese acceder a cualquier información contenida en una base de datos, llamando a las bases de datos móviles la **nueva generación**, donde un dispositivo móvil puede conectarse mediante servidores a redes de comunicación digitales inalámbricas.

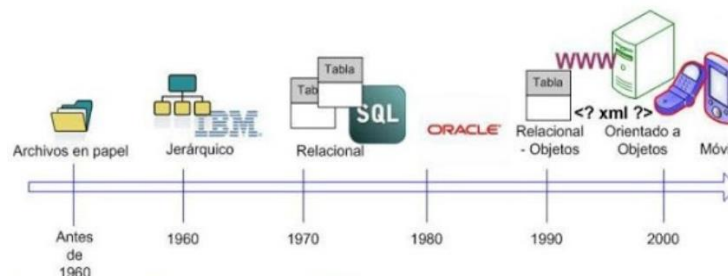


Fig. 1. Desarrollo histórico de los SMBD

Abrir conexiones

Para conectarse a Microsoft SQL Server, use el objeto SqlConnection del proveedor de datos .NET Framework para SQL Server. Para conectarse a un origen de datos OLE DB, use el objeto OleDbConnection del proveedor de datos .NET Framework para OLE DB. Para conectarse a un origen de datos ODBC, utilice el objeto OdbcConnection del proveedor de datos .NET Framework para ODBC. Para conectarse a un origen de datos Oracle, utilice el objeto OracleConnection del proveedor de datos .NET Framework para ODBC. Para almacenar de forma segura y recuperar cadenas de conexión, vea proteger la información de conexión.

Cierre de conexiones

Recomendamos cerrar siempre la conexión cuando termine de utilizarla, para que la conexión pueda regresar al grupo. El bloque Using de Visual Basic o C# elimina automáticamente la conexión cuando el código sale del bloque, incluso en el caso de una excepción no controlada.

También puede utilizar los métodos Close o Dispose del objeto de conexión correspondiente al proveedor que esté utilizando. Es posible que las conexiones que no se cierran explícitamente no se puedan agregar ni puedan regresar al grupo. Por ejemplo, una conexión

que se ha salido del ámbito pero que no se ha cerrado explícitamente solo se devolverá al grupo de conexión si se ha alcanzado el tamaño máximo del grupo y la conexión aún es válida.

La conexión a un servidor de bases de datos suele constar de varios pasos que requieren mucho tiempo. Se debe establecer un canal físico, como un socket o una canalización con nombre, debe tener lugar el protocolo de enlace con el servidor, se debe analizar la información de la cadena de conexión, el servidor debe autenticar la conexión, se deben ejecutar comprobaciones para la inscripción en la transacción actual, etc.

En la práctica, la mayoría de las aplicaciones solamente utilizan unas cuantas configuraciones diferentes para las conexiones. Esto significa que durante la ejecución de la aplicación, muchas conexiones idénticas se abrirán y cerrarán de forma repetida. Para minimizar el costo de abrir conexiones, ADO.NET emplea una técnica de optimización llamada *agrupación de conexiones*.

La agrupación de conexiones reduce el número de veces que es necesario abrir nuevas conexiones. El *agrupador* mantiene la propiedad de la conexión física. Para administrar las conexiones, mantiene un conjunto de conexiones activas para cada configuración de conexión dada. Cada vez que un usuario llama a Open en una conexión, el agrupador comprueba si hay una conexión disponible en el grupo. Si hay disponible una conexión agrupada, la devuelve a la persona que llama en lugar de abrir una nueva. Cuando la aplicación llama a Close en la conexión, el agrupador la devuelve al conjunto agrupado de conexiones activas en lugar de cerrarla. Una vez que la conexión vuelve al grupo, ya está preparada para volverse a utilizar en la siguiente llamada a Open.

Solo se pueden agrupar conexiones con la misma configuración. ADO.NET mantiene varios grupos de forma simultánea, uno para cada configuración. Las conexiones se dividen en grupos por cadena de conexión, y por identidad de Windows si se utiliza seguridad integrada. Las conexiones también se agrupan en función de si están incluidas en una transacción. Cuando se usa ChangePassword, la instancia de SqlCredential afecta al grupo de conexiones. Distintas instancias de SqlCredential usarán diferentes grupos de conexiones, incluso si el identificador de usuario y la contraseña son iguales.

Conclusión

Hoy en día el desarrollo del software ha llegado a tal nivel en el que las aplicaciones necesitan ser más potentes para satisfacer la demanda de información, que debe ser segura y constante pero no de mucho serviría tenerla ordenada en tablas relacionadas a otras simplemente almacenada, lo que hay que hacer es darle uso necesitamos poder modificarla y tener acceso constante para eliminaciones o inserciones nuevas, ado.net nos permite conectarnos con el gestor de la base de datos para poder obtener los beneficios del acceso a la información(datos).

Referencias

Partesddel.com, obtenido de:

<https://www.partesdel.com/resumen.html>

Microsoft. (2009). conexión a base de datos. 04/dic/2018, de Microsoft Sitio web:
<https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/establishing-the-connection>

Microsoft. (2009). conexión a base de datos. 04/dic/2018, de Microsoft Sitio web:
<https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/sql-server-connection-pooling>

Referencia:

Quiroz, J. (2012). ADO . diciembre 04, 2018, de SlideShare Sitio web:
<https://es.slideshare.net/jairoUNT/ado-15334882>

<http://caes1987.galeon.com/index8.htm>