CHƯƠNG 5

ADO.NET

Nội dung

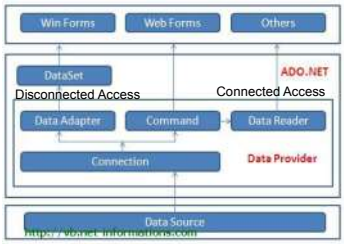
1. Giới thiệu về ADO.NET
2. Các thành phần của ADO.NET
3. Thao tác với dữ liệu bằng kiến trúc kết nối
4. Mô hình 3 tầng

1. Giới thiệu ADO.NET

- **ADO.NET** (ActiveX Data Objects .NET) là một thành phần trong **.NET Framework** đảm nhiệm vai **trò thao tác với CSDL**

1. Giới thiệu ADO.NET

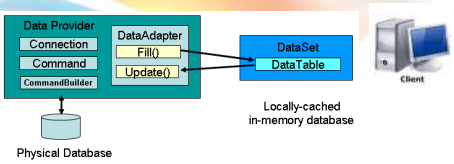
- ADO.NET hoạt động theo cả hai kiến trúc kết nối và không kết nối



Kiến trúc ADO.NET

2. Các thành phần chính của ADO.NET

- ADO gồm hai phần
  - .NET data provider: cung cấp các lớp để **kết nối**, **thực thi lệnh SQL** trên CSDL và **lấy kết quả trả về**
  - Dataset: lưu trữ **bản sao** của CSDL trên bộ nhớ. Cung cấp các phương thức cho phép ứng dụng thao tác với bản sao



2. Các thành phần chính của ADO.NET

Data provider cung cấp các lớp để tương tác với CSDL

- Data provider gồm 4 thành phần chính
  - **Connection**: kết nối với CSDL
  - **Command**: thực thi các lệnh sql để lấy dữ liệu từ CSDL hoặc thay đổi CSDL
  - **DataReader**: đọc dữ liệu tuần tự từ CSDL
  - **DataAdapter**: lấy dữ liệu từ CSDL lưu vào dataset và cập nhật CSDL

2. Các thành phần chính của ADO.NET

- Các data provider được đặt trong các namespace khác nhau

Provider	Namespace
SQL Server	System.Data.SqlClient
OLE DB	System.Data.OleDb
ODBC	System.Data.Odbc
Oracle	System.Data.OracleClient

- Các lớp của data provider

Object	SQL Server	OLE DB	ODBC	Oracle
Connection	SqlConnection	OleDbConnection	OdbcConnection	OracleConnection
Command	SqlCommand	OleDbCommand	OdbcCommand	OracleCommand
Data reader	SqlDataReader	OldeDbDataReader	OdbcDataReader	OracleDataReader
Data adapter	SqlDataAdapter	OleDbDataAdapter	OdbcDataAdapter	OracleDataAdapter

3. Thao tác với dữ liệu bằng kiến trúc kết nối

1. Tạo, mở Connection
2. Tạo Command
3. Thực thi Command để truy vấn | cập nhật dữ liệu
4. Đóng connection

8

3.1. SqlConnection

Thực hiện **kết nối** với CSDL

- **Phương thức khởi tạo**

  new SqlConnection(connectionString);

- **Thuộc tính và phương thức**

Thuộc tính/ Phương thức	Mô tả
ConnectionString	Cung cấp thông tin để truy cập đến CSDL SQL Server
Open()	Mở kết nối
Close()	Đóng kết nối

9

3.1. SqlConnection

```
//Khai báo connection
SqlConnection conn;
//Tạo connection
string connString = "data source=localhost;
initial catalog=QuanLyBanHang,
integrated security=true";
conn = new SqlConnection(connString);
//Mở connection
conn.Open();
```

Tên Server CSDL

Tên CSDL

Đăng nhập bằng quyền Windows

10

3.2. SqlCommand

- **Thực thi** các lệnh sql để lấy dữ liệu từ CSDL hoặc thay đổi CSDL
- **Phương thức khởi tạo**

  new SqlCommand(cmdText, connection);

  - cmdText: chuỗi lệnh sql truy vấn, cập nhật dữ liệu
  - connection: đối tượng SqlConnection

11

3.2. SqlCommand

- **ExecuteReader**(): trả lại đối tượng DataReader của data provider
- **ExecuteNonQuery**() – được sử dụng để thực thi các câu lệnh Insert, Update, Delete SQL, trả lại số dòng bị tác động
- **ExecuteScalar**() – Trả lại một giá trị duy nhất, ví dụ hàm SQL SUM.

12

3.2. SqlCommand – ví dụ

```
//câu lệnh xóa bản ghi
string strDelete = "DELETE tblKhachHang WHERE
MaKH='" + txtMa.Text + "'";
//Khai báo và khởi tạo command
SqlCommand cmd = new SqlCommand(strDelete,
conn);
//Thực thi câu lệnh delete
cmd.ExecuteNonQuery();
```

13

3.2. SqlCommand – ví dụ

```
//Câu lệnh lấy các bản ghi trong bảng khách hàng
string sqlSelect = "SELECT * FROM tblKhachHang";
//Khai báo và tạo Command
SqlCommand cmd =
        new SqlCommand(sqlSelect, conn);
//thực thi câu lệnh SELECT
SqlDataReader dr;
dr = cmd.ExecuteReader();
```

14

Sử dụng command có tham số

Ví dụ lọc dữ liệu trong truy vấn

```
string sqlSelect = "SELECT * FROM tblKhachHang
WHERE MaKH='" + txtMa.Text + "'";
SqlCommand cmd = new SqlCommand(sqlSelect,
conn);
```

Sử dụng command có tham số

B1: Xây dựng command text có tham số:

```
string sqlSelect = "SELECT * FROM tblKhachHang
WHERE MaKH=@makh";
```

B2. Sau khi khởi tạo command, gán các giá trị thích hợp cho mỗi tham số của SqlCommand

```
SqlCommand cmd = new SqlCommand(sqlSelect,
conn);
cmd.Parameters.AddWithValue("makh", txtMa.Text);
```

3.4. SqlDataReader

- DataReader **đọc** dữ liệu tuần tự từ CSDL
- Đọc dữ liệu từ DataReader
  - Gọi phương thức **Read** cho mỗi bản ghi. Phương thức trả lại false khi không còn bản ghi.
  - Truy cập các trường. Tham số là tên trường hoặc vị trí của trường

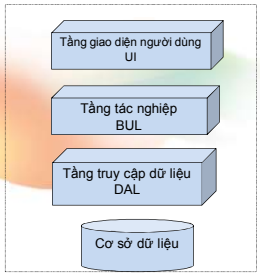
```
while (myReader.Read())
{
  str += myReader[1];
  str += myReader["field"];
}
```

  - Đóng DataReader
  - Đóng Connection

17

Ứng dụng 3 tầng

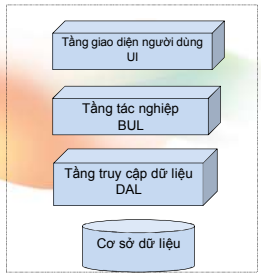
- **Tầng truy cập DL:** thực hiện kết nối, truy vấn, cập nhật trực tiếp cơ sở dữ liệu, (là lớp gần nhất với CSDL - có thể có mô tả dữ liệu)
- **Tầng tác nghiệp**: Mô tả cụ thể xử lý dữ liệu, chứa tất cả các logic của chương trình. Bất cứ khi nào người dùng muốn cập nhật các logic của chương trình chỉ cần cập nhật lớp này.



Ứng dụng 3 tầng

- **Tầng giao diện:** đảm nhiệm tương tác với người sử dụng

  Chú ý: Nên tạo mỗi layer là 1 project



Ứng dụng 3 tầng