Khoa Công nghệ Thông tin

Chương 2:
TRUY CẬP CƠ SỞ DỮ LIỆU VỚI ADO.NET

Mục tiêu bài học

1. Nắm vững kiến trúc ADO.NET
2. Hiểu rõ các đối tượng ADO.NET
3. Kết nối CSDL
4. Thực hiện được các thao tác cập nhật dữ liệu

Nội dung chương 2

Khoa Công nghệ Thông tin

1. Giới thiệu ADO.NET
2. Các đối tượng trong ADO.NET

1. Giới thiệu ADO.NET

1.1. Đặc điểm – lợi ích

1.2. Mô hình các đối tượng

1.3. Các bước truy xuất cơ sở dữ liệu

1.4. Kết nối CSDL

Giới thiệu ADO.NET

Khoa Công nghệ Thông tin

- ADO.NET là một tập hợp các class cho phép truy cập, thao tác với CSDL
- Viết tắt của ActiveX Data Object.NET (ADO.NET) phát triển từ ADO VB6.0
- ADO.NET cung cấp phương pháp truy xuất với nhiều loại dữ liệu khác nhau: SQL Server, OLEDB, XML
- ADO.NET là một kiến trúc dữ liệu rời rạc, làm việc theo mô hình ngắt kết nối => giảm lưu lượng truy cập vào CSDL cùng lúc, tiết kiệm tài nguyên.

Giới thiệu ADO.NET

Khoa Công nghệ Thông tin

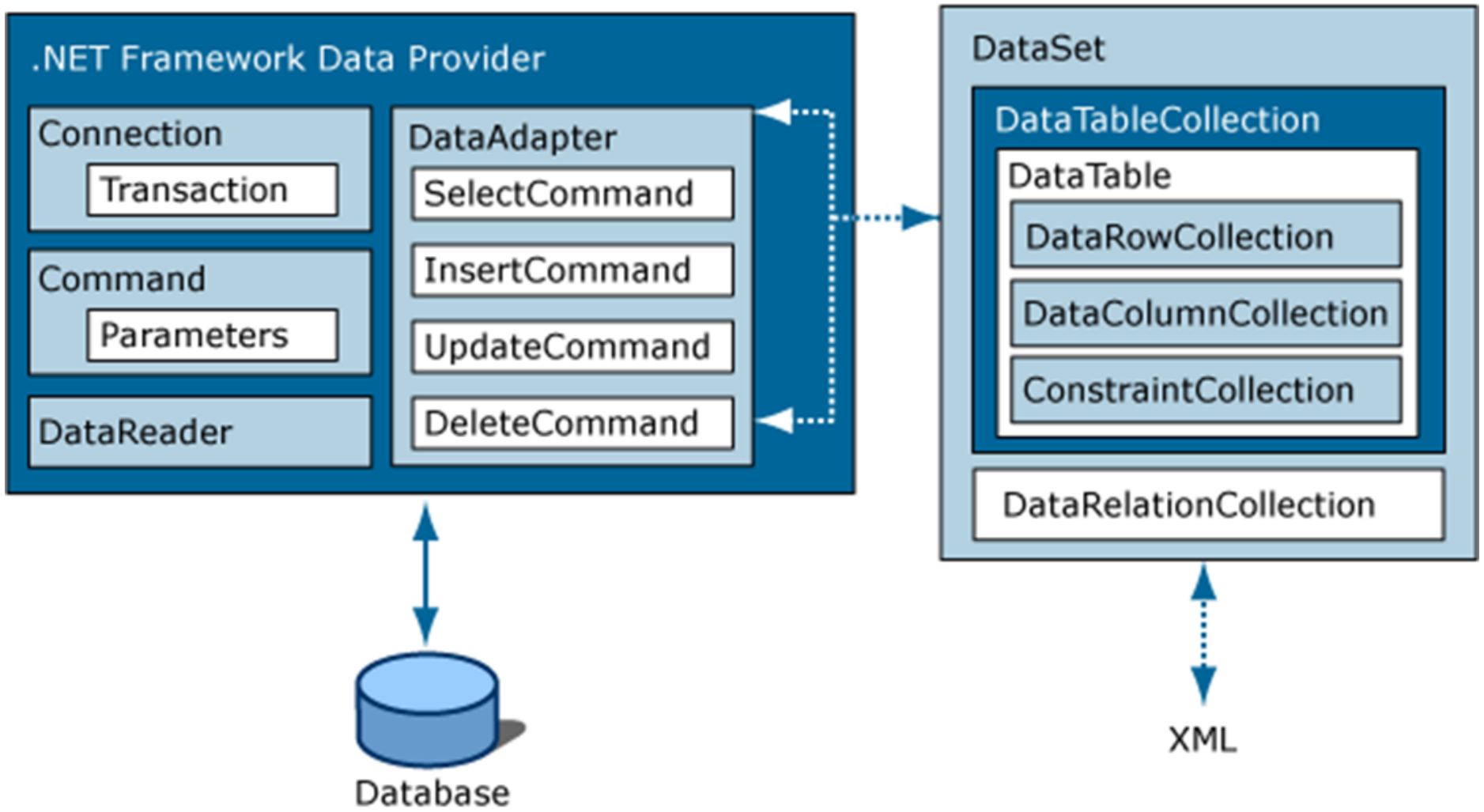
- ADO.NET sử dụng các trình cung cấp dữ liệu .NET (.NET Data Providers) để liên kết ứng dụng với nguồn dữ liệu
- .NET Framework hỗ trợ 2 loại Data Providers
  - SQL Server Data Provider – chỉ làm việc với HQTCSDL SQL Server, có namespace System.Data.SqlClient – SQL Server 7 trở lên
  - OLEDB Data Provider – làm việc với tất cả OleDB Provider như: SQL OleDB Provider, Oracle OleDB Provider, Jet OleDB Provider, có namespace System.Data.Oledb

So sánh SQL Server .NET Data Provider và the OLE DB .NET Data Provider

1.1. Lợi ích của ADO.NET

- Làm việc theo mô hình CSDL ngắt kết nối
- Được .Net Framework hỗ trợ sẵn nên dễ dàng xây dựng và triển khai ứng dụng.
- Hỗ trợ chuyển đổi giữa mô hình CSDL quan hệ và định dạng dữ liệu XML.

1.2. Mô hình các đối tượng trong ADO.NET



1.3. Các bước truy xuất CSDL

Khoa Công nghệ Thông tin

Các bước truy xuất CSDL:

1. Chuẩn bị:
   a. ConnectionString
   b. Truy vấn dữ liệu
2. Khai báo các đối tượng ADO.NET cần thiết:
   a. Connection
   b. Command
   c. …
3. Thiết lập các thông số cho Command
4. Mở kết nối
5. Thực thi truy vấn
6. [Xử lý kết quả truy vấn]
7. Đóng, hủy các đối tượng

1.4. Kết nối CSDL

Khoa Công nghệ Thông tin

- Kết nối là một component được sử dụng để thiết lập một kết nối đến CSDL từ 1 Datasource.
- Có 2 kiểu kết nối: SQLConnection và OleDBConnection

Tên	Mô tả
ConnectionString	Cung cấp thông tin như data source, tên cơ sở dữ liệu, được sử dụng để thiết lập kết nối với một CSDL
Open()	Mở một kết nối với data source được khai báo tại ConnectionString
Close()	Được sử dụng để đóng kết nối với data source
State	Được sử dụng để kiểm tra trạng thái của một kết nối. 0: kết nối đang đóng 1: kết nối đang mở

2. Các đối tượng trong ADO.NET

2.1. Connection

2.2. Command

2.3. Parameter

2.4. DataReader

2.5. DataAdapter

2.6. DataTable, DataColumn, DataRow

2.7. DataView

2.8. DataSet

2.9. DataRelation

2.1. Connection

Connection: dùng để kết nối với nguồn dữ liệu

1. Khai báo: SqlConnection <tênOBJ> = **new** SqlConnection([connectionString]);
2. Thuộc tính:

- ConnectionString: truyền vào chuỗi kết nối tới nguồn dữ liệu
- State: trạng thái hiện kết nối: Open, Closed, Connecting, Fetching, Broken

Chuỗi ConnectionString

Thông số kết nối CSDL Sql Server gồm:

Đối với chế độ: Windows Authentication

Data Source = <tên server cài DB|local>;
Initial Catalog= <tên DB>;
Integrated Security=True|SSPI;

Đối với chế độ: SQL Server Authentication

Data Source = <tên server cài DB|local>;
Initial Catalog = <tên DB>;
User ID =<tên truy cập>;
Password = <mật khẩu>;

Có thể thay:

- Data Source ~ Server
- Initial Catalog ~ Database
- User ID ~ UID
- Password ~ PWD

2.1. Connection

3. Phương thức

- Open(): mở kết nối
- Close(): đóng kết nối
- CreateCommand(): **SqlCommand**: tạo ra một đối tượng Command
- BeginTransaction(): **SqlTransaction** tạo ra đối tượng Transaction
- Dispose(): hủy đối tượng
  - có thể sử dụng cấu trúc using(<khai báo đối tượng>){….} để .NET tự hủy đối tượng ngay sau khi thực thi xong

Sử dụng cấu trúc using khai báo dữ liệu


```
using (SqlConnection sqlConnection = new SqlConnection(str))
{
        using (SqlCommand sqlCommand = sqlConnection.CreateCommand())
            {
                        sqlConnection .Open();
                        //các thao tác dữ liệu
                        sqlConnection .Close();
             }
}
```

2.2. Command

Command: dùng để thực hiện các truy vấn với CSDL

1. Khởi tạo:
   - SqlCommand <tênObj> = new SqlCommand(commandText, connection);
   - SqlCommand <tênObj> = <đối tượng Sqlconnection>.CreateCommand();

2.2. Command

Khoa Công nghệ Thông tin

2. Thuộc tính:
   a. Connection: truyền vào đối tượng Connection
   b. CommandType: chỉ định loại truy vấn sẽ thực thi (StoredProcedure|Text|TableDirect)
   c. CommandText: truyền vào câu lệnh SQL hoặc tên Stored procedure
   d. Parameters: tập hợp các tham số
      i. Add(<Parameter>)
      ii. AddWithValue(name, value)
      iii. Clear();

2.2. Command

Khoa Công nghệ Thông tin

3. Phương thức:
   a. Phương thức ExecuteReader(): **DataReader** : thực thi với câu lệnh SQL Select và trả về luồng dữ liệu qua đối tượng DataReader
   b. Phương thức ExecuteScalar(): **Object** : thực thi câu lệnh SQL Select và trả về một giá trị ở hàng đầu tiên và cột đầu tiên.
   c. Phương thức ExecuteNonQuery(): **int** : thực thi câu lệnh NonQuery như Select … Into…, Insert, Update, Delete …

Bài tập vận dụng

Khoa Công nghệ Thông tin

Thêm dữ liệu vào bảng sinh viên sử dụng câu lệnh SQL trực tiếp

ADMIN.TT88_ver2 - dbo.tblSINHVIEN

Column Name	Data Type	Allow Nulls
iMaSV	int	☐
sHoTen	nvarchar(50)	☑
dNgaySinh	date	☑
sDiaChi	nvarchar(255)	☑
sSoDienThoai	varchar(30)	☑
bGioiTinh	bit	☑
		☐

1. Viết hàm ThemSinhVien() bao gồm: câu lệnh truy vấn; tạo Connection; tạo Command; thực thi truy vấn; đóng kết nối
2. Hàm Main(): tạo Connection; nhập dữ liệu; thực hiện thêm dữ liệu – gọi hàm – đóng đối tượng

2.3. Parameter

Parameter: dùng để truyền tham số cho truy vấn.

1. Khởi tạo:
   - SqlParameter <tênObj> = new SqlParameter();
   - <Command Obj>.Parameters.Add(
     parameterName: string
     , sqlDbType: SqlDbType
     , size: int
     , sourceColumn: string);

2.3. Parameter

Khoa Công nghệ Thông tin

2. Thuộc tính:

- ParameterName
- DbType: SQLDbTypes.Int, .BigInt, .VarChar, ….
- Size
- ParameterDirection: .Input|Output|InputOutput|Return Value
- Value
- SourceColumn

3. Phương thức:

- Add()
- AddWithValue(name, value)

Bài tập vận dụng

Khoa Công nghệ Thông tin

Thêm dữ liệu vào bảng sinh viên sử dụng Stored Procedure

ADMIN.TT88_ver2 - dbo.tblSINHVIEN

Column Name	Data Type	Allow Nulls
iMaSV	int	☐
sHoTen	nvarchar(50)	☑
dNgaySinh	date	☑
sDiaChi	nvarchar(255)	☑
sSoDienThoai	varchar(30)	☑
bGioiTinh	bit	☑
		☐

```
CREATE PROC Insert_tblSINHVIEN
@maSV int,
@tenSV nvarchar(50),
@ngaySinh date,
@diaChi nvarchar(255),
@soDienThoai varchar(30),
@gioiTinh bit
AS
INSERT INTO tblSINHVIEN(iMaSV, sHoTen, dNgaySinh, sDiaChi, sSoDienThoai, sGioiTinh)
VALUES(@maSV, @tenSV, @ngaySinh, @diaChi, @soDienThoai, @gioiTinh)
```

Thực thi Stored Procedure trong lập trình C#

Stored Procedure trong SQL Server [2]**:**

- Một stored procedure (truy vấn chứa sẵn) là một hoặc nhiều câu lệnh SQL được lưu trữ trong CSDL dưới dạng một đối tượng có thể thực thi
- Stored procedure có thể được gọi thông qua chương trình ứng dụng, qua các stored procedures khác và qua các triggers
- Có thể tạo và sử dụng các Parameters (tham số) để truyền dữ liệu vào/ra stored procedure nhằm tăng khả năng tùy biến của chúng

Thực thi Stored Procedure trong lập trình C#

Khoa Công nghệ Thông tin

Tại sao phải sử dụng Stored Procedure?

- Cho phép Module hóa việc truy xuất dữ liệu
- Cho phép phân cấp truy cập theo tính năng đến các bảng dữ liệu
- Giảm tải lưu thông qua mạng
  - Lời gọi
  - Xử lý tại DBMS trước khi trả kết quả
- Thực thi nhanh hơn, an toàn bảo mật hơn.
- Giảm thiểu khả năng gây lỗi so với khi người dùng tương tác trực tiếp với bảng dữ liệu
- Có thể áp dụng các Transaction để đảm bảo tính toàn vẹn

Thực thi Stored Procedure trong lập trình C#

Tạo và sử dụng Stored Procedure trong SQL Server:

- Tạo:

```
CREATE PROCEDURE <procedure_name>
[@<parm_name> datatype][OUTPUT]
[,....]
AS
        SQL Statements
```

- Thực thi:

```
EXECUTE <procedure_name> [@<parm_name> = value][,...][OUTPUT]
```

Thực thi Stored Procedure trong lập trình C#

Khoa Công nghệ Thông tin

Thực thi Stored Procedure trong C#:

```
using (SqlConnection sqlConnection = new SqlConnection(str))
{
        using (SqlCommand sqlCommand = sqlConnection.CreateCommand())
            {
                sqlCommand.CommandType = CommandType.StoredProcedure;
                sqlCommand.CommandText = "procedure_name";

                //thiết lập các parameters
                //…
                sqlConnection .Open();
                //thực thi truy vấn
                sqlCommand.Execute…();
                //[xử lý kết quả truy vấn]
                sqlConnection .Close();
            }
}
```

2.4. DataReader

DataReader: Dùng để duyệt dữ liệu lấy ra từ CSDL

- Mỗi thời điểm (Read()), DataReader chỉ làm việc với 1 bản ghi và chỉ duyệt các bản ghi theo chiều tiến -> tốc độ nhanh
- Trong thời gian tồn tại, DataReader khóa cứng 1 đối tượng DataConnection với nó (-> dùng xong phải Close DataReader để giải phóng Connection)
- Khai báo: **SqlDataReader** <tênOBJ> = <Command>.**ExecuteReader()**;

2.4. DataReader

Khoa Công nghệ Thông tin

- Thuộc tính:
  - HasRows: true/false -> có bản ghi trong reader hay không
  - FieldCount: số trường trong reader
  - IsClosed: true/false -> reader đã bị đóng chưa
- Phương thức:
  - Read(): true/false -> đọc 1 bản ghi từ nguồn dữ liệu vào reader, nếu đọc được sẽ trả ra true, ngược lại trả ra false
  - Close(): đóng reader
  - Get<type>(i): đọc dữ liệu trường thứ i theo kiểu <type>

  Có thể truy xuất tới 1 trường trong reader bằng cách <tên reader>[“tên trường”]

Bài tập vận dụng

Hiện danh sách sinh viên sử dụng Stored Procedure và DataReader

ADMIN.TT88_ver2 - dbo.tblSINHVIEN

	Column Name	Data Type	Allow Nulls
🔑	iMaSV	int	☐
	sHoTen	nvarchar(50)	☑
	dNgaySinh	date	☑
	sDiaChi	nvarchar(255)	☑
	sSoDienThoai	varchar(30)	☑
	bGioiTinh	bit	☑
			☐

```
CREATE PROC Select_tblSINHVIEN
AS
SELECT iMaSV, sHoTen, dNgaySinh, sDiaChi, sSoDienThoai, sGioiTinh
FROM tblSINHVIEN
```

Bài tập vận dụng

Khoa Công nghệ Thông tin

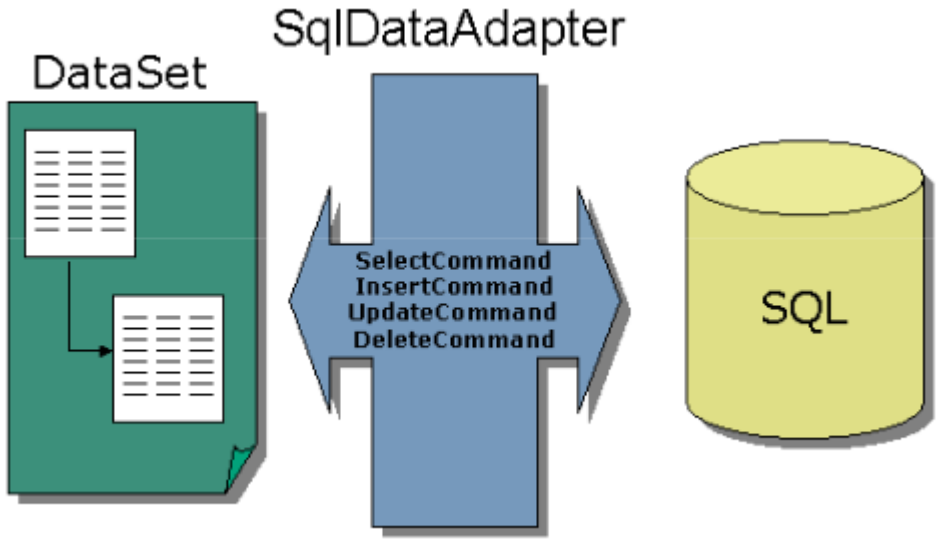
Viết hàm HienSinhVien()

```
public static void HienSinhVien(string connectionString)
{
    string queryInsert = "Select_tblSINHVIEN";
    using (SqlConnection sqlConnetion = new SqlConnection(connectionString))
    {
        using (SqlCommand sqlCommand = sqlConnetion.CreateCommand())
        {
            sqlCommand.CommandText = queryInsert;
            sqlCommand.CommandType = CommandType.StoredProcedure;
            sqlConnetion.Open();
            using(SqlDataReader dataReader = sqlCommand.ExecuteReader())
            {
                if (dataReader.HasRows)
                {
                    while (dataReader.Read())
                    {
                        Console.WriteLine("{0}\t{1}\t{2}\t{3}",
                                          dataReader["iMaSV"]
                                        , dataReader["sHoTen"]
                                        , dataReader["dNgaySinh"]
                                        , dataReader["sGioiTinh"]);
                    }
                }
            }
            sqlConnetion.Close();
        }
    }
}
```

2.5. DataAdapter [4]

DataAdapter: Dùng để thực hiện việc trao đổi dữ liệu từ DBMS tới DataSet và ngược lại

1. Khai báo: **SqlDataAdapter** <tênObj> = **new SqlDataAdapter**([sqlCommand]);
2. Thuộc tính:
   - ContinueUpdateOnError: True/False
   - SelectCommand
   - InsertCommand
   - UpdateCommand
   - DeleteCommand

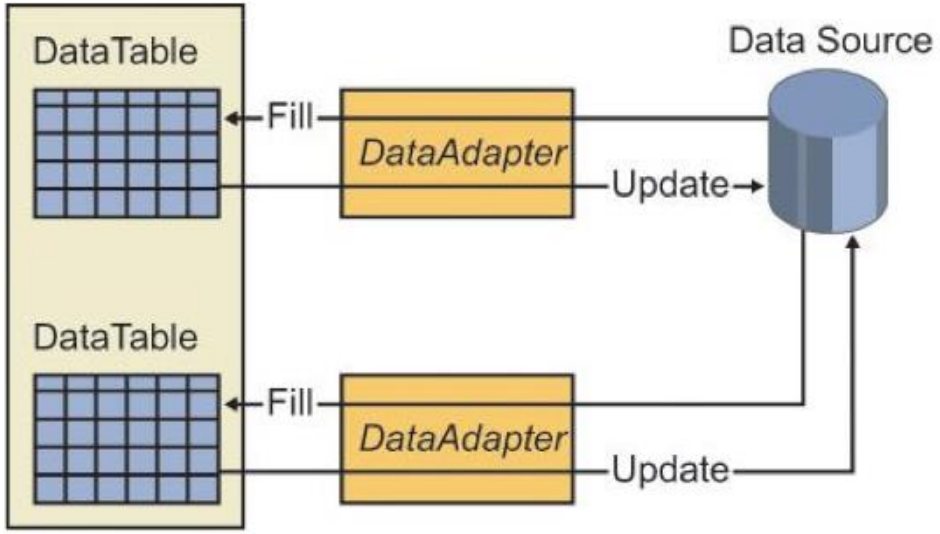


2.5. DataAdapter

Khoa Công nghệ Thông tin

3. Phương thức:

- **Fill**(DataSet|DataTable):
  sử dụng SelectCommand đưa dữ liệu vào DataSet|DataTable
- **Update**(DataSet|DataTable):
  cập nhật các thay đổi dữ liệu (InsertCommand, UpdateCommand, DeleteCommand) từ DataSet|DataTable về CSDL



2.6. DataTable

DataTable: thể hiện một bảng dữ liệu trong bộ nhớ

1. Khai báo:

    DataTable <tên obj> = **new DataTable**([tên table])

2.6. DataTable

Khoa Công nghệ Thông tin

2. Thuộc tính:

- TableName
- Columns
  - Add(DataColumn)
- Rows
  - Add(DataRow)
- HasErrors
- DefaultView

3. Phương thức:

- NewRow(): DataRow
- GetChanges([RowState]): DataTable
- AcceptChange()
- RejectChange()
- Select([filter] [,order] [,DataViewState]): DataRow[]

2.6. DataColumn

Khoa Công nghệ Thông tin

DataColumn: Đại diện 1 trường của DataTable

1. Khai báo:

DataColumn <tên obj> = **new** DataColumn ([ColumnName] [, DataType] [, Expression]);

2.6. DataColumn

Khoa Công nghệ Thông tin

2. Thuộc tính:

a. ColumnName
b. DataType = System.Type.GetType(<tên kiểu .NET>)
c. DefaultValue
d. Expression
e. MaxLength
f. Table
g. Unique: true/false

2.6. DataRow

Khoa Công nghệ Thông tin

DataRow: Đại diện 1 dòng của DataTable

1. Khai báo: DataRow <tên obj> = **new** DataRow ();

   Lưu ý: tạo mới 1 dòng: DataTable.NewRow();

2.6. DataRow

Khoa Công nghệ Thông tin

2. Thuộc tính:
   a. Table
   b. RowState
   c. HasErrors: true/false
   d. <DataRow>[tên trường]: Truy nhập giá trị 1 trường

3. Phương thức:
   a. Delete()
   b. GetChildRows(DataRelation|RelationName): DataRow[]

2.7. DataView

Khoa Công nghệ Thông tin

- **DataView: Thể hiện cách nhìn khác (View) của dữ liệu trong DataTable**
- DataView có thể:
  - Kết gán dữ liệu (DataBind) với các điều khiển
  - Sắp xếp, lọc, tìm kiếm và duyệt các bản ghi.
- Khai báo: DataView <tênObj>= **new** DataView([DataTable])

2.7. DataView

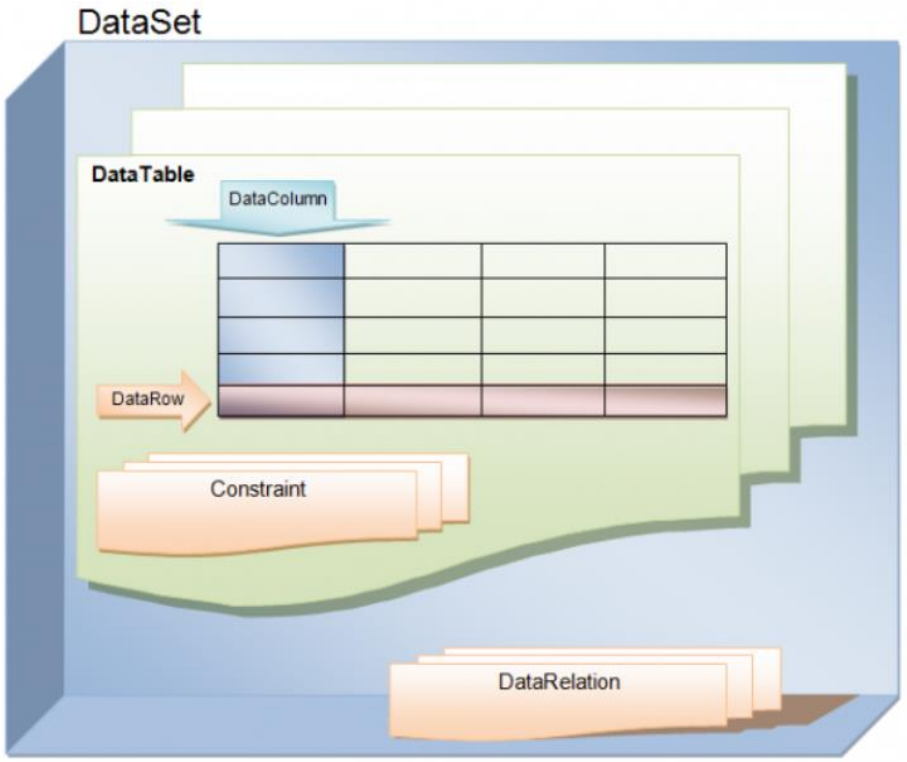
Khoa Công nghệ Thông tin

- Thuộc tính:
  - Table
  - Sort
  - RowFilter = <điều kiện lọc bản ghi>
  - RowStateFilter
  - DataViewRowState: Added|CurrentRows|Deleted| ModifiedCurrent| ModifiedOriginal|None|OriginalRows|Unchanged

2.8. DataSet

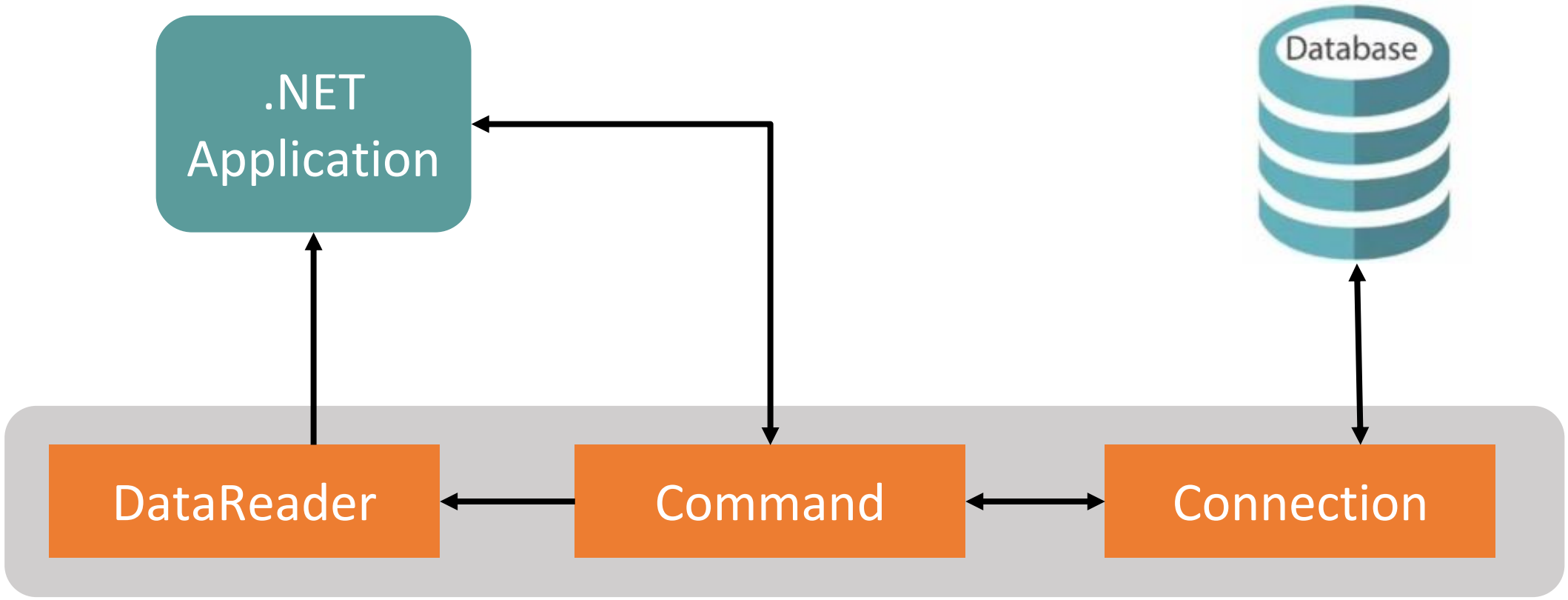
Khoa Công nghệ Thông tin

- DataSet: Là đối tượng quan trọng trong ADO.NET
- Mô hình tương tự một DB trong bộ nhớ
  - Có thể chứa nhiều DataTable.
  - Giữa các DataTable có thể thiết lập quan hệ bằng DataRelation
- DataSet chỉ chứa dữ liệu, không giữ kết nối
  - Lấy dữ liệu từ DB vào DataSet bằng Fill() của DataAdapter
  - Có thể cập nhật dữ liệu các DataTable của DataSet, sau đó mới cập nhật trở lại DB bằng Update() của DataAdapter

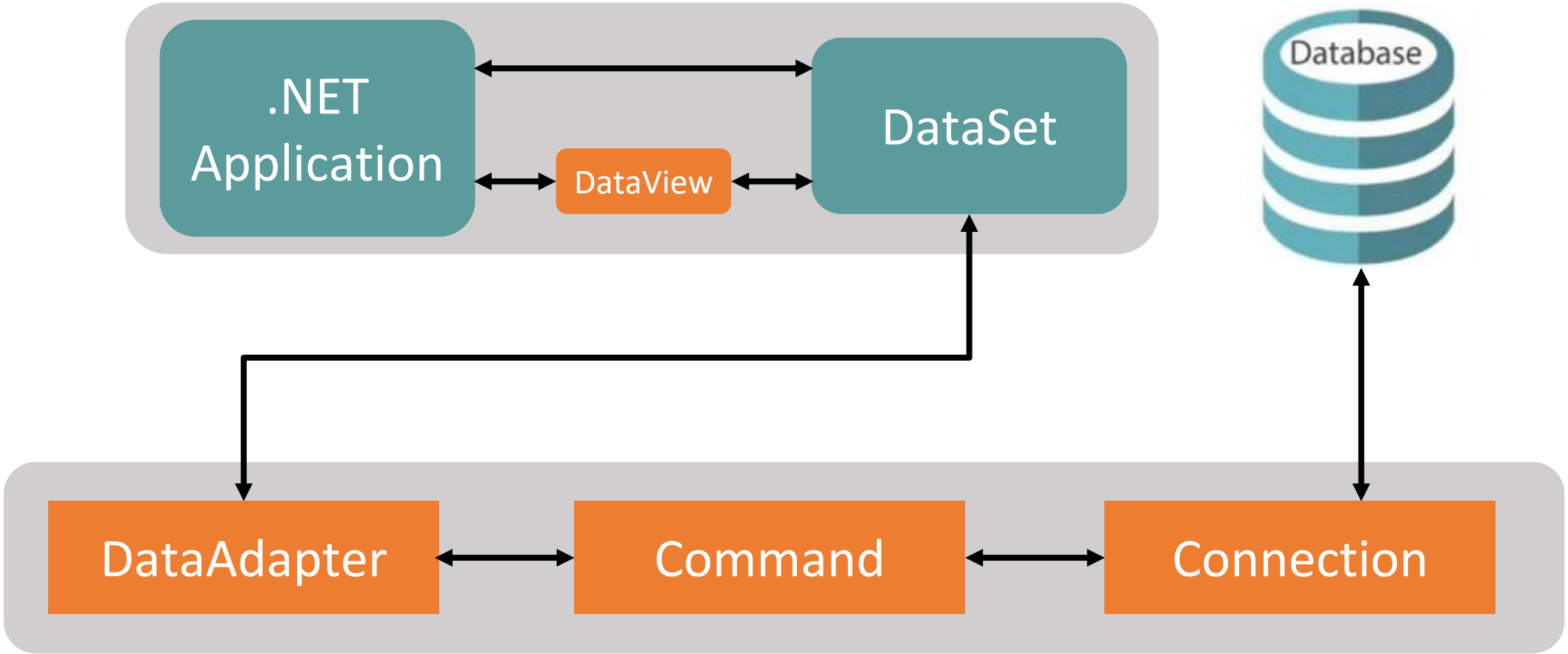


Cấu trúc của DataSet

ADO.NET Connection Oriented Architecture



ADO.NET Disconnected Architecture



2.8. DataSet

Khoa Công nghệ Thông tin

1. Khai báo: DataSet <tênObj> = **new** DataSet()
2. Thuộc tính:
   a. Tables
   b. Relations
   c. DataSetName
   d. HasErrors: true/false

2.8. DataSet

Khoa Công nghệ Thông tin

3. Phương thức
    a. HasChanges([DataRowState]): true/false
    b. GetChanges([DataRowState])
    c. AcceptChange()
    d. RejectChange()
    e. GetXml()
    f. WriteXml(…)
    g. ReadXml(<tên file>)

2.9. DataRelation

Khoa Công nghệ Thông tin

DataRelation: Để thiết lập quan hệ giữa các DataTable trong DataSet

1. Khai báo:

DataRelation <tênObj> = **new** DataRelation(name, parentColumn, childColumn);

2. Thuộc tính:

a. Relationname
b. ParentTable, ChildTable
c. ParentColumn, ChildColumn
d. ParentKeyConstraint, ChildKeyConstraint

Bài tập vận dụng

Khoa Công nghệ Thông tin

Sử dụng DataAdapter làm việc theo mô hình ngắt kết nối.

1. Hiện toàn bộ danh sách sinh viên
2. Hiện danh sách sinh viên yêu cầu chỉ lấy sinh viên nữ và sắp xếp theo ngày sinh sử dụng DataView
3. Kiểm tra sự tồn tại của khóa chính.
4. Làm các chức năng Thêm, Sửa, Xóa sinh viên.


THE END