

## Chương 2

# TỔNG QUAN VỀ ADO.NET

- Giúp sinh viên nắm được:
  - Kiến thức cơ sở về các lớp của ADO.Net
  - Cách viết một chương trình đơn giản có kết nối và đọc các dòng từ cơ sở dữ liệu.
  - Cách xây dựng ứng dụng Windows Form

- ADO.Net là gì?
- Phân loại các lớp của ADO.Net
- Các lớp trong nhóm kết nối
  - Connection, Command
  - Parameter, ParameterCollection
  - DataReader, DataAdapter
  - ...
- Các lớp trong nhóm không kết nối
  - DataSet, DataTable, DataView
  - DataColumn, DataRow
  - DataRelation, Constraint

○ **ADO.NET** là một công nghệ - công cụ có thể truy xuất dữ liệu theo 2 phương thức:

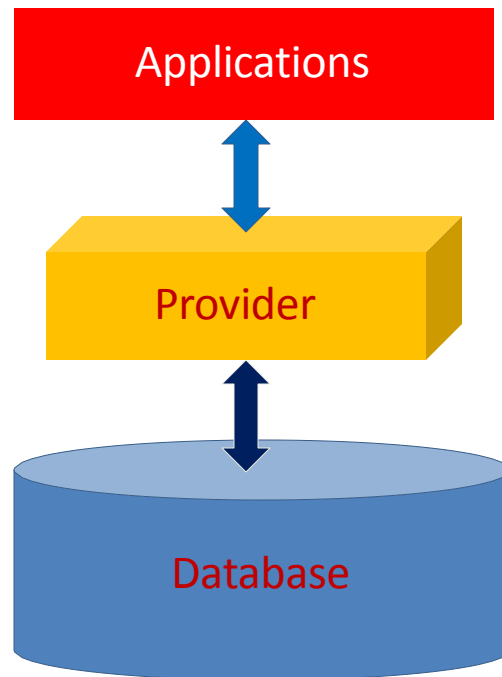
- Kết nối: Tương tác trực tiếp với một cơ sở dữ liệu thông qua đối tượng thuộc các lớp kết nối (hay trình điều khiển được quản lý - **managed provider**).
- Không kết nối: thông tin từ cơ sở dữ liệu được lưu trực tiếp trong bộ nhớ máy tính mà chương trình đang chạy

○ Các đối tượng này cho phép kết nối tới cơ sở dữ liệu và thực thi các lệnh SQL khi luồng kết nối được mở.

○ ADO.Net phân phối dữ liệu dựa trên **chuẩn XML**. Vì thế, nó có thể truyền dữ liệu từ nơi này sang nơi khác qua Internet và có thể **xuyên qua các tường lửa**.

○ Mục đích của ADO.Net là hỗ trợ xây dựng các ứng dụng **N-Tier** hoặc có thể chạy trên nhiều server.

- Data Provider (hay Driver – trình điều khiển) là một thuật ngữ ám chỉ một thư viện nhị phân - được viết dưới dạng các hàm API (giao diện lập trình ứng dụng).



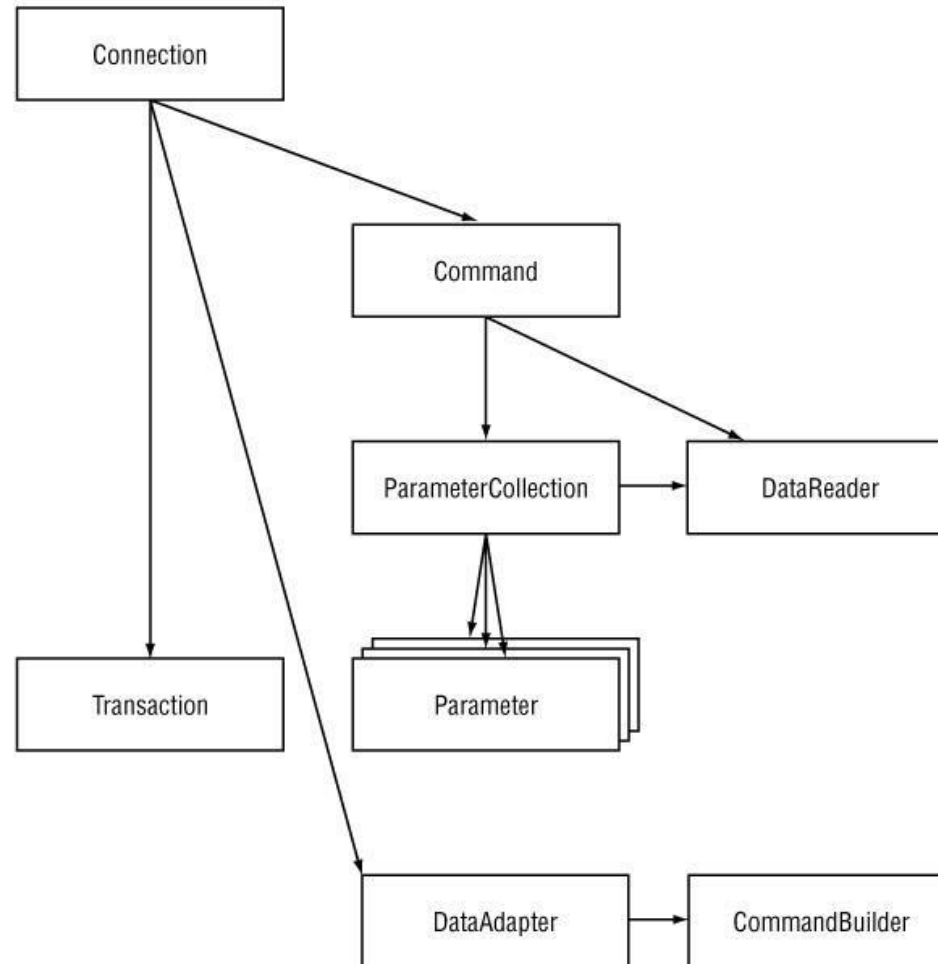
## ○ Các lớp kết nối (**managed provider**)

- Tạo kết nối trực tiếp
- Đồng bộ hóa dữ liệu được lưu trên máy cục bộ với csdl.
- Đọc các dòng dữ liệu từ cơ sở dữ liệu theo một hướng.
- Sử dụng tùy thuộc vào loại cơ sở dữ liệu muốn kết nối tới.

## ○ Các lớp không kết nối (**generic data**)

- Lưu trữ bản sao thông tin lấy được từ cơ sở dữ liệu.
- Không phụ thuộc vào loại cơ sở dữ liệu mà bạn dùng, nghĩa là các lớp này được dùng chung cho dữ liệu được lấy từ cơ sở dữ liệu SQL Server, Access hay Oracle...
- Các lớp này biểu diễn thông tin lấy từ cơ sở dữ liệu ở định dạng XML.

Managed Provider Objects



- **SQL Server Managed Provider Classes:** được dùng để kết nối tới một cơ sở dữ liệu SQL Server.
- **OLE DB Managed Provider Classes:** được dùng để kết nối tới bất kỳ cơ sở dữ liệu nào hỗ trợ chuẩn OLE DB (*Object Linking and Embedding for Databases*) như Access hay Oracle.
- **ODBC Managed Provider Classes:** được dùng để kết nối tới các cơ sở dữ liệu có hỗ trợ chuẩn ODBC (*Open Database Connectivity*).
  - Tốc độ truy xuất chậm → chỉ nên dùng nhóm lớp này khi không còn cách nào khác để thay thế nhóm OLE DB.



- Tất cả các nhóm lớp kết nối này đều được xây dựng cùng một tập chức năng, nghĩa là có cùng các hàm xử lý.
- Tên của các chuẩn này được đặt trước tên của các lớp kết nối để cho biết nhóm lớp nào đang được dùng.
- Chẳng hạn, nếu tên lớp kết nối bắt đầu bằng ***Sql*** thì cơ sở dữ liệu đang dùng được quản lý bởi SQL Server.
- Vì thế, để đơn giản, ta chỉ gọi tên chung của chúng. Chẳng hạn, lớp Connection, lớp Command...

- Muốn giao tiếp với một cơ sở dữ liệu, trước hết phải thực hiện một kết nối tới cơ sở dữ liệu thông qua lớp **Connection**.
- Có 3 lớp kết nối:
  - SqlConnection
  - OleDbConnection
  - OdbcConnection
- Mở kết nối trước khi thực thi lệnh bằng phương thức **Open()** và đóng kết nối bằng phương thức **Close()**

## ○Đối tượng **Command** được dùng để

- thực thi một lệnh SQL như SELECT, INSERT, UPDATE hay DELETE.
- Gọi một thủ tục hay lấy các dòng dữ liệu từ một bảng cụ thể.

## ○Đối tượng **Command** sử dụng một đối tượng **Connection** và yêu cầu kết nối này phải được mở trước khi thực thi truy vấn.

## ○Có 3 lớp Command

- SqlCommand
- OleDbCommand
- OdbcCommand

- Đối tượng **Parameter** được dùng để gửi một tham số vào trong đối tượng *Command*.
- Nó được dùng để lưu giá trị tham số được gửi vào trong một thủ tục (Stored Procedure) hay một lệnh SQL.
- Khi có nhiều tham số, chúng được lưu và truyền vào đối tượng *Command* thông qua đối tượng **ParameterCollection**.
- Có 3 lớp tham số:
  - SqlParameter
  - OleDbParameter
  - OdbcParameter

- Đối tượng thuộc lớp này dùng để đọc các dòng dữ liệu được từ cơ sở dữ liệu bởi đối tượng **Command**.
- Đối tượng **DataReader** chỉ được dùng để đọc dữ liệu theo một chiều từ đầu tới cuối và không thể cập nhật dữ liệu cho các dòng trong cơ sở dữ liệu.
- **DataReader** có thể dùng để thay thế cho DataSet và việc đọc dữ liệu từ DataReader thường nhanh hơn đọc dữ liệu từ **DataSet**.
- Có 3 lớp DataReader:
  - SqlDataReader
  - OleDbDataReader
  - OdbcDataReader

## ○Đối tượng **DataAdapter** được dùng để

- chuyển các dòng trong một cơ sở dữ liệu vào một DataSet
- đồng bộ dữ liệu đã thay đổi ở máy cục bộ tới cơ sở dữ liệu.
- Việc đồng bộ được thực hiện qua một đối tượng Connection.

## ○Có thể đọc các dòng trong cơ sở dữ liệu vào một **DataSet** qua một DataAdapter, thay đổi dữ liệu đó. Sau đó, cập nhật các thay đổi lên cơ sở dữ liệu thông qua đối tượng **Connection**.

## ○Có 3 lớp DataAdapter:

- SqlDataAdapter
- OleDbDataAdapter
- OdbcDataAdapter

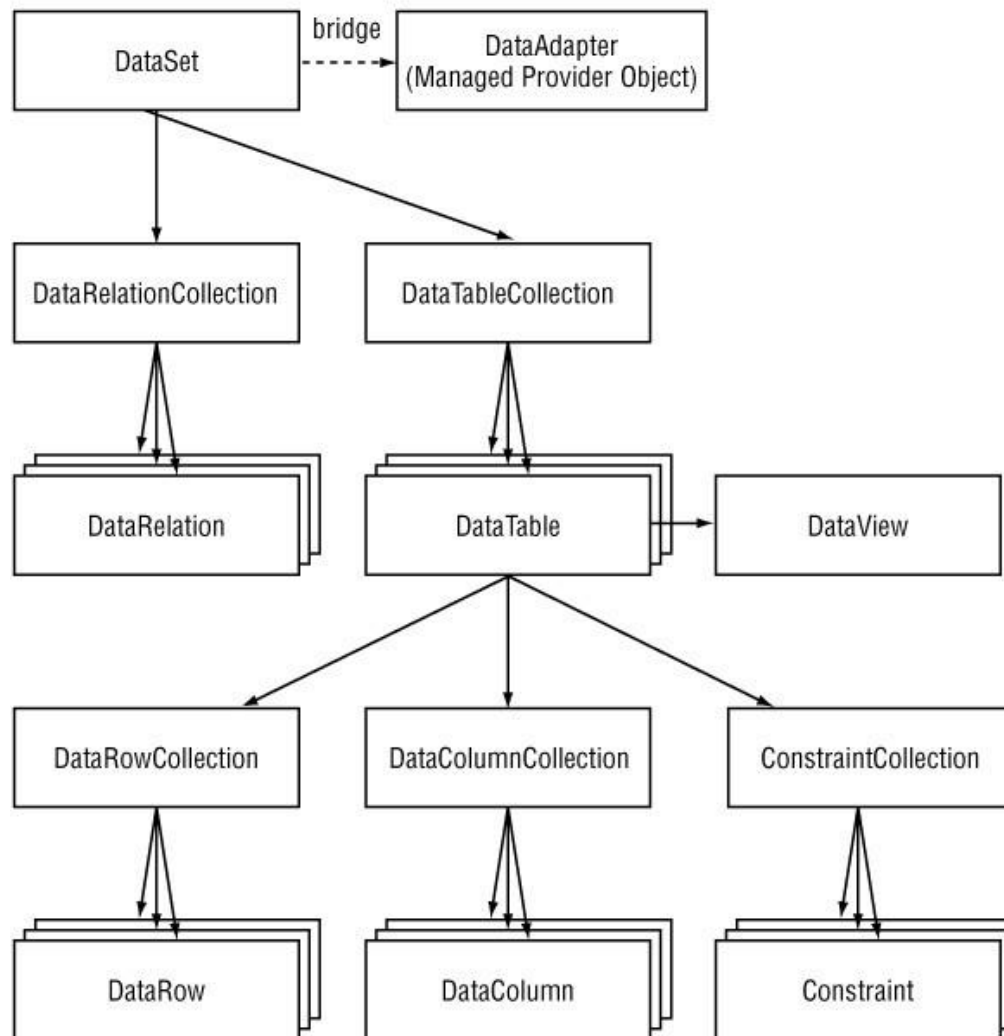
- Có 3 lớp **CommandBuilder**:
  - SqlCommandBuilder, OleDbCommandBuilder, OdbcCommandBuilder
- Đối tượng **CommandBuilder** tự động sinh ra các lệnh INSERT, UPDATE, DELETE trên một bảng và đồng bộ dữ liệu bị thay đổi trong DataSet lên cơ sở dữ liệu.
- Đối tượng **Transaction** được dùng để biểu diễn một giao dịch cơ sở dữ liệu (database transaction).
- Một giao dịch cơ sở dữ liệu là một nhóm các lệnh làm thay đổi các dòng trong cơ sở dữ liệu.
- Các lệnh lệnh này được xem như một đơn vị công việc về mặt logic.
- Có 3 lớp giao dịch (*transaction*):
  - SqlTransaction, OleDbTransaction và OdbcTransaction.

- Các lớp kết nối sử dụng cho SQL Server được khai báo trong namespace *System.Data.SqlClient*.
- Các lớp kết nối cho cơ sở dữ liệu hỗ trợ OLE DB được khai báo trong namespace *System.Data.OleDb*.
- Các lớp kết nối cho cơ sở dữ liệu hỗ trợ ODBC được khai báo trong namespace *System.Data.Odbc*.



- Các đối tượng không kết nối (hay các đối tượng chứa dữ liệu) cho phép
  - Lưu trữ một bản sao thông tin lấy từ cơ sở dữ liệu.
  - Khi đã ngắt kết nối tới cơ sở dữ liệu.
  - Đọc các dòng theo thứ tự bất kỳ
  - Tìm kiếm, sắp xếp hay trích lọc các dòng một cách linh hoạt.
  - Tạo ra các thay đổi trên dữ liệu, sau đó đồng bộ (cập nhật) các thay đổi này vào cơ sở dữ liệu.
- Cầu nối giữa các lớp kết nối và các lớp không kết nối là DataAdapter.

Generic Data Set Objects



- Đối tượng của lớp **DataSet** được dùng để biểu diễn một bản sao thông tin được lưu trong cơ sở dữ liệu.
- Một **DataSet** có thể biểu diễn các cấu trúc như một cơ sở dữ liệu bao gồm các bảng, các dòng và cột, các ràng buộc như tính duy nhất (*unique*) hoặc ràng buộc về khóa ngoại.
- Đối tượng **DataSet** cũng được dùng để chứa dữ liệu có định dạng XML.

- Đối tượng của lớp *DataTable* dùng để biểu diễn một bảng.
- Một *DataSet* có thể chứa nhiều bảng hay nhiều đối tượng *DataTable*.
- Những bảng này được truy xuất qua thuộc tính ***Tables*** của lớp DataSet. Thuộc tính này có kiểu ***DataTableCollection***.
- Một *DataTable* có thể chứa nhiều dòng, nhiều cột (tương ứng là ***DataRow*** và ***DataColumn***).

- Đối tượng có kiểu *DataRow* dùng để biểu diễn một hàng (một dòng trong DataTable).
- Đối tượng thuộc lớp *DataColumn* được dùng để biểu diễn một cột của bảng.
- Một *DataTable* có thể chứa nhiều *DataRow* và được truy xuất thông qua thuộc tính *Rows* của lớp DataTable. Thuộc tính này có kiểu *DataRowCollection*.
- Một *DataTable* cũng có thể chứa nhiều *DataColumn* và được truy xuất thông qua thuộc tính *Columns* của lớp DataTable. Thuộc tính này có kiểu *DataColumnCollection*.

- Đối tượng thuộc kiểu *Constraint* dùng để biểu diễn một ràng buộc cơ sở dữ liệu áp dụng cho một hoặc nhiều đối tượng *DataColumn* của một bảng (*DataTable*).
- Một *DataTable* có thể lưu trữ nhiều đối tượng *Constraint* và được truy xuất thông qua thuộc tính *Constraints* của lớp *DataTable*. Thuộc tính này có kiểu là *ConstraintCollection*.
- Có 2 loại ràng buộc:
  - *UniqueConstraint*:
  - *ForeignKeyConstraint*:

- Đối tượng có kiểu *DataView* được dùng để xem nội dung dữ liệu của các dòng trong một *DataTable* bằng cách dùng một bộ lọc với điều kiện nào đó.
- Đối tượng *DataRelation* dùng để biểu diễn một quan hệ giữa hai đối tượng *DataTable*.
- Có thể dùng một đối tượng *DataRelation* làm quan hệ cha-con giữa hai bảng như trong cơ sở dữ liệu.
- Một *DataSet* có thể chứa nhiều đối tượng *DataRelation*. Các quan hệ này được truy xuất qua thuộc tính *Relations* của lớp *DataSet*. Thuộc tính này có kiểu *DataRelationCollection*.

- Các lớp *DataSet*, *DataTable*, *DataRow*, *DataColumn*, *DataRelation*, *Constraint* và *DataView* được khai báo trong namespace *System.Data*.



- Tạo chuỗi kết nối
- Tạo đối tượng kết nối tới cơ sở dữ liệu (**Connection**)
- Tạo lệnh truy vấn
- Tạo đối tượng thực thi lệnh (**Command**)
- Thiết lập lệnh truy vấn
- Tạo đối tượng điều phối dữ liệu (**DataAdapter**)
- Tạo đối tượng lưu trữ (**DataSet / DataTable**)
- Mở kết nối tới cơ sở dữ liệu
- Gọi phương thức lấy dữ liệu
- Đóng kết nối
- Lấy dữ liệu từ DataSet / DataTable
- Hiển thị dữ liệu

- Console Application
- Windows Application

**HẾT CHƯƠNG 2**