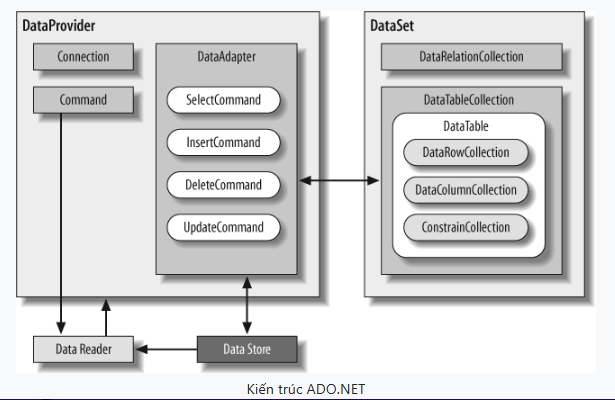
**DataAdapter, DataSet, và DataTable**

****

|  |
| --- |
| **DataTable**   * **Vai trò:** **DataTable** là một bảng dữ liệu trong bộ nhớ, tương tự một bảng trong cơ sở dữ liệu. * **Đặc điểm:**   + Lưu trữ dữ liệu dạng hàng và cột.   + Có thể tồn tại độc lập hoặc là một phần của **DataSet**.   + Cung cấp các phương thức để thêm, xóa, hoặc cập nhật dữ liệu trong bảng. |

|  |
| --- |
| **DataSet**   * **Vai trò**: **DataSet** là một tập hợp các bảng dữ liệu (**DataTable**) được lưu trữ trong bộ nhớ. * **Đặc điểm:**   + Có thể chứa nhiều bảng (**DataTable**), mối quan hệ giữa các bảng (**DataRelation**), và các ràng buộc.   + Không phụ thuộc vào nguồn dữ liệu, làm việc hoàn toàn trong bộ nhớ.   + Thích hợp cho các ứng dụng xử lý dữ liệu phức tạp, ngoại tuyến. |

|  |
| --- |
| **DataAdapter**  **DataAdapter** là một thành phần của ADO.NET, được sử dụng để làm cầu nối giữa một nguồn dữ liệu  (ví dụ: Database) và một đối tượng trong bộ nhớ (như ***DataSet*** hoặc **DataTable**). Nó cung cấp cơ chế để:   * Lấy dữ liệu từ cơ sở dữ liệu và điền vào đối tượng trong bộ nhớ. * Đồng bộ hóa các thay đổi từ đối tượng trong bộ nhớ trở lại cơ sở dữ liệu.   **-------------------------------------------------------------------------------------------------------------------** Cách hoạt động của DataAdapter **DataAdapter** hoạt động như một bộ điều phối với các phương thức chính:   1. **Fill**:    * Lấy dữ liệu từ cơ sở dữ liệu đổ vào **DataSet** hoặc **DataTable**.    * Thực hiện câu truy vấn SELECT. 2. **Update**:    * Cập nhật các thay đổi từ **DataSet** hoặc **DataTable** trở lại cơ sở dữ liệu.    * Yêu cầu phải cấu hình đúng các lệnh INSERT, UPDATE, DELETE.   **DataAdapter** thường sử dụng cùng với các lệnh SQL thông qua một đối tượng **Command.** Ví dụ:   * **SelectCommand**: Lệnh SELECT để lấy dữ liệu. * **InsertCommand**: Lệnh INSERT để thêm dữ liệu. * **UpdateCommand**: Lệnh UPDATE để cập nhật dữ liệu. * **DeleteCommand**: Lệnh DELETE để xóa dữ liệu.   **< Chú ý > :**Trong C# với **SqlDataAdapter**, mỗi **SqlDataAdapter** chỉ hỗ trợ một bảng trong DataSet tại một thời điểm. Điều này có nghĩa là **mỗi SqlDataAdapter chỉ có thể quản lý một bảng duy nhất** (ví dụ: Doctors, Patients, etc.).  **Bạn không thể trực tiếp cấu hình một SqlDataAdapter để xử lý nhiều bảng trong DataSet cùng một lúc**. |

|  |
| --- |
| **Nhược điểm**   * Hiệu suất có thể thấp nếu làm việc với lượng dữ liệu lớn, do dữ liệu được giữ trong bộ nhớ. * Dễ xảy ra lỗi đồng bộ khi dữ liệu trong cơ sở dữ liệu thay đổi đồng thời bởi nhiều người dùng.   Nếu bạn chỉ cần xử lý dữ liệu nhanh và ít, **SqlDataAdapter** là một lựa chọn tốt. Nếu yêu cầu cao hơn, bạn nên cân nhắc sử dụng Entity Framework hoặc các công cụ ORM khác. |

|  |
| --- |
| **Cách hoạt động của SourceVersion:**  **SourceVersion** là một thuộc tính của đối tượng SqlParameter trong ADO.NET. Nó giúp xác định **dữ liệu nào trong DataRow** sẽ được sử dụng khi tham số được gán vào câu lệnh SQL.   * **DataRowVersion** có ba giá trị chính:   + **Current**: Giá trị hiện tại của cột trong DataRow (giá trị đã thay đổi).   + **Original**: Giá trị gốc của cột trong DataRow (giá trị chưa thay đổi).   + **Proposed**: Giá trị được đề xuất cho cột (nếu có).   Khi bạn thực hiện các thay đổi trong DataTable, ADO.NET sẽ theo dõi các phiên bản khác nhau của mỗi cột trong DataRow. Các thay đổi này có thể bao gồm việc thêm, sửa, hoặc xóa dữ liệu. **SourceVersion** xác định phiên bản của dữ liệu nào sẽ được dùng trong câu lệnh SQL.  Khi bạn thay đổi dữ liệu của một dòng trong **DataTable**, giá trị hiện tại trong bảng đó **đã thay đổi ngay lập tức**. Tuy nhiên, ADO.NET vẫn giữ lại một bản sao của giá trị ban đầu (giá trị trước khi thay đổi) để bạn có thể sử dụng khi cần, nhờ vào **DataRowVersion**. |