**DTO là gì và để làm gì?**

**DTO (Data Transfer Object)** là một lớp trung gian được sử dụng để truyền dữ liệu giữa các tầng trong ứng dụng (ví dụ: từ tầng dữ liệu - DAO - đến tầng giao diện - UI). Nó không chứa logic nghiệp vụ, mà chỉ đóng vai trò như một "hộp chứa" dữ liệu đơn giản.

**Mục đích của DTO**

1. **Tách biệt dữ liệu và logic**:
   * DTO giúp tách dữ liệu từ CSDL (hoặc DAO) khỏi các đối tượng nghiệp vụ phức tạp, tránh việc truyền trực tiếp DataTable hoặc DataRow lên tầng UI.
2. **Kiểm soát dữ liệu truyền lên UI**:
   * Bạn có thể chọn lọc những trường cần thiết từ bảng CSDL để gửi lên giao diện, thay vì gửi toàn bộ dữ liệu thô.
3. **Tối ưu hóa truyền dữ liệu**:
   * Trong các ứng dụng lớn hoặc phân tán, DTO giúp giảm lượng dữ liệu truyền qua mạng (dù với WinForms điều này ít quan trọng hơn).
4. **Dễ dàng ánh xạ (mapping)**:
   * DTO cho phép ánh xạ dữ liệu từ CSDL thành các đối tượng C# dễ sử dụng, sau đó hiển thị trên các control như DataGridView, ListBox, v.v.

**Ví dụ**

* Nếu bạn lấy dữ liệu từ bảng KhachHang bằng DataProvider.ExecuteQuery, kết quả là một DataTable. Để hiển thị trong DataGridView, bạn có thể dùng trực tiếp DataTable. Nhưng nếu muốn xử lý dữ liệu (lọc, sắp xếp) hoặc truyền giữa các lớp, một DTO như KhachHangDTO sẽ tiện lợi hơn.

**Có cần tạo DTO trong dự án này không?**

Dựa trên dự án của bạn (WinForms với SQL Server và DataProvider), việc tạo DTO không phải là bắt buộc, nhưng nó **được khuyến khích** trong các trường hợp sau:

1. **Nếu bạn muốn code rõ ràng và dễ bảo trì**:
   * Thay vì làm việc trực tiếp với DataTable hoặc DataRow, DTO giúp bạn định nghĩa các lớp với thuộc tính rõ ràng (ví dụ: KhachHangDTO với SoCCCD, HoTen, v.v.). Điều này làm code dễ đọc và dễ mở rộng hơn.
2. **Nếu bạn cần xử lý dữ liệu trước khi hiển thị**:
   * Ví dụ: Ghép họ tên khách hàng với số CCCD thành một chuỗi để hiển thị, hoặc tính toán tổng chi phí từ dữ liệu thô.
3. **Nếu dự án có thể mở rộng sau này**:
   * Khi thêm tầng nghiệp vụ (Business Layer) hoặc chuyển sang mô hình phân tán (ví dụ: Web API), DTO sẽ giúp bạn dễ dàng tái cấu trúc mà không phải sửa lại toàn bộ code.
4. **Nếu bạn chỉ dùng DataTable đơn giản**:
   * Với WinForms, bạn có thể dùng trực tiếp DataTable để gán vào DataGridView.DataSource mà không cần DTO. Tuy nhiên, điều này có thể làm code kém linh hoạt khi cần xử lý dữ liệu phức tạp.

**Kết luận**

* **Không bắt buộc**: Nếu bạn chỉ cần hiển thị dữ liệu trực tiếp từ CSDL lên UI và không xử lý nhiều logic, thì cứ dùng DataTable từ DataProvider.
* **Nên tạo DTO**: Nếu bạn muốn code chuyên nghiệp, dễ bảo trì, và chuẩn bị cho mở rộng sau này, hãy tạo DTO cho từng bảng.