



# THIẾT KẾ THỰC THỂ NGHIỆP VỤ

*(Designing Business Entities)*



TS. Huỳnh Hữu Nghĩa

huynhhuunghia@iuh.edu.vn

# Nội dung:

---

- Khái quát
- B1: Chọn lựa trình bày.
- B2: Chọn một thiết kế cho các thực thể nghiệp vụ.
- B3: Xác định hỗ trợ tuần tự hóa.

# Khái quát

- Các thực thể nghiệp vụ lưu trữ giá trị dữ liệu và tìm kiếm thông qua các thuộc tính, chứa và quản lý dữ liệu nghiệp vụ được dùng bởi ứng dụng và cung cấp quyền truy cập chương trình trạng thái đầy đủ dữ liệu nghiệp vụ và tính năng có liên quan.
- Thực thể nghiệp vụ cũng cần xác thực dữ liệu chứa trong thực thể và đóng gói logic nghiệp vụ để đảm bảo tính nhất quán và triển khai các quy tắc nghiệp vụ và hình vi.

# Khái quát

- Thiết kế hay lựa chọn các thực thể nghiệp vụ phù hợp là cực kỳ quan trọng để tối đa hóa hiệu suất và hiệu quả của lớp nghiệp vụ.
- Các nội dung:
  - ✓ Tìm kiếm các định dạng dữ liệu khác nhau và cách mà dữ liệu sẽ sử dụng trong ứng dụng,
  - ✓ Nắm cách định dạng dữ liệu từ việc xác định các quy tắc nghiệp vụ có thể triển khai trong thiết kế,
  - ✓ Hiểu về các tùy chọn thiết kế cho đối tượng tùy chỉnh,
  - ✓ Cách hỗ trợ tuần tự hóa với các định dạng dữ liệu khác nhau.

# Bước 1: Lựa chọn trình bày

- Các đối tượng nghiệp vụ tùy chỉnh (Custom Business Objects):
  - Nó mô tả các thực thể trong hệ thống, sử dụng phù hợp khi muốn đóng gói các quy tắc nghiệp vụ phức tạp hay hành vi cùng với dữ liệu liên quan.
  - Kỹ thuật ánh xạ đối tượng/quan hệ (O/RM) như ADO.NET Entity Framework (EF) hay Nhibernate có thể sử dụng để tạo đối tượng.

# Bước 1: Lựa chọn trình bày

- Các đối tượng nghiệp vụ tùy chỉnh (Custom Business Objects):
  - Việc truy cập các đối tượng nghiệp vụ tùy chỉnh được thông qua AppDomain (xử lý hay ranh giới vật lý) có thể triển khai một lớp dịch vụ cung cấp truy cập thông qua DTO (Data Transfer Objects) và các hoạt động cập nhật hay chỉnh sửa các đối tượng nghiệp vụ tùy chỉnh.

# Bước 1: Lựa chọn trình bày

## ➤ DataSet hay DataTable:

- Là một dạng cơ sở dữ liệu trong bộ nhớ thường ánh xạ sát với lược đồ cơ sở dữ liệu.
- Chỉ dùng nếu không sử dụng cơ chế O/RM và đang xây dựng ứng dụng hướng dữ liệu trong đó dữ liệu ở các bảng phân phối logic ứng dụng rất gần với lược đồ cơ sở dữ liệu.

# Bước 1: Lựa chọn trình bày

## ➤ DataSet hay DataTable:

- Không thể được mở rộng để đóng gói logic nghiệp vụ hoặc các quy tắc nghiệp vụ.
- Mặc dù, Nó có thể tuần tự hóa đến XML nhưng không nên hiển thị qua các ranh giới dịch vụ hay quy trình.



# Bước 1: Lựa chọn trình bày

## ➤ XML:

- Là một định dạng dựa trên chuẩn chứa dữ liệu có cấu trúc, thường được sử dụng để biểu diễn các thực thể nghiệp vụ chỉ khi lớp trình bày yêu cầu hoặc nếu logic phải làm việc với nội dung dựa trên lược đồ của nó.
- Ví dụ, một hệ thống định tuyến tin nhắn trong đó logic định tuyến các tin nhắn dựa trên một số nút biết được trong tài liệu XML

# Bước 2: Thiết kế Thực thể nghiệp vụ

- Hướng tiếp cận thiết kế được sử dụng cho các đối tượng tùy chỉnh là dựa trên loại đối tượng dự kiến sử dụng.
- Ví dụ, *các thực thể mô hình lĩnh vực yêu cầu phân tích chuyên sâu một lĩnh vực nghiệp vụ, trong khi các thực thể chuẩn hóa bảng yêu cầu hiểu lược đồ cơ sở dữ liệu.*

# Bước 2: Thiết kế Thực thể nghiệp vụ

- **Mô hình lĩnh vực** là một mẫu thiết kế hướng đối tượng:
  - Mục tiêu là xác định các đối tượng nghiệp vụ biểu diễn các thực thể ở thế giới thực trong lĩnh vực nghiệp vụ.
  - Khi thiết kế mô hình lĩnh vực, các thực thể lĩnh vực hay nghiệp vụ chứa cả hành vi và cấu trúc. *(ie. Các mối quan hệ và quy tắc nghiệp vụ được đóng gói bên trong mô hình lĩnh vực)*

# Bước 2: Thiết kế Thực thể nghiệp vụ

- **Mô hình lĩnh vực** là một mẫu thiết kế hướng đối tượng:
  - Thiết kế mô hình lĩnh vực yêu cầu phân tích chuyên sâu lĩnh vực nghiệp vụ và thường không nên ánh xạ đến các mô hình quan hệ được sử dụng hầu hết các CSDL.

# Bước 2: Thiết kế Thực thể nghiệp vụ

- **Mô hình lĩnh vực** là một mẫu thiết kế hướng đối tượng:
  - Thiết kế DM khi có các quy tắc nghiệp vụ phức tạp liên quan đến lĩnh vực nghiệp vụ, thiết kế rich client và lĩnh vực nghiệp vụ có thể được khởi tạo và lưu trữ trong bộ nhớ, hay đang làm việc với một lớp nghiệp vụ stateless yêu cầu việc khởi tạo mô hình lĩnh vực với mọi yêu cầu.

# Bước 2: Thiết kế Thực thể nghiệp vụ

- **Tiêu chuẩn bảng** là một mẫu thiết kế hướng đối tượng:
  - Mục tiêu là xác định các thực thể dựa trên tables hay views trong CSDL.
  - Các hoạt động sử dụng để truy cập CSDL và lưu vào các thực thể tiêu chuẩn bảng thường được đóng gói bên trong thực thể.

# Bước 2: Thiết kế Thực thể nghiệp vụ

- **Tiêu chuẩn bảng** là một mẫu thiết kế hướng đối tượng:
  - Tuy nhiên, có thể sử dụng các thành phần truy cập dữ liệu để thực hiện các hoạt động cơ sở dữ liệu và lưu vào các thực thể tiêu chuẩn bảng.
  - Sử dụng tiêu chuẩn bảng nếu tables hoặc views trong CSDL thể hiện chắc chắn các thực thể nghiệp vụ được dùng bởi ứng dụng hay logic nghiệp vụ và các hoạt động liên quan đến một table hay view đơn lẻ.

# Bước 2: Thiết kế Thực thể nghiệp vụ

## ➤ Các Đối tượng XML tùy chỉnh:

- Biểu diễn dữ liệu XML bị khử tuần tự hóa được dùng trong mã ứng dụng. Các đối tượng được khởi tạo từ các lớp được xác định với các thuộc tính bên trong cấu trúc XML.
- Microsoft .NET Framework cung cấp các thành phần được sử dụng để khử tuần tự hóa dữ liệu XML thành các đối tượng và các đối tượng tuần tự hóa thành XML



# Bước 2: Thiết kế Thực thể nghiệp vụ

## ➤ Các Đối tượng XML tùy chỉnh:

- Sử dụng các đối tượng XML tùy chỉnh nếu dữ liệu đang tiêu thụ có sẵn trong định dạng XML (*ie. Các file XML hay các hoạt động CSDL trả về kết quả XML*);
- Cần sinh ra dữ liệu XML từ các nguồn dữ liệu không XML hoặc đang làm việc với dữ liệu dựa trên tài liệu chỉ đọc.

# Bước 2: Thiết kế Thực thể nghiệp vụ

- Khi sử dụng các đối tượng tùy chỉnh, các thực thể nghiệp vụ không bắt buộc tuân theo cùng thiết kế.
- Ví dụ, một khía cạnh ứng dụng có các quy tắc phức tạp có thể yêu cầu thiết kế DM, phần còn lại sử dụng các đối tượng XML, một thiết kế Table Module hay các đối tượng lĩnh vực nếu phù hợp.

# Bước 3: Xác định hỗ trợ tuần tự hóa

- Bạn phải xác định cách chuyển các thực thể nghiệp vụ qua các ranh giới.
- Để chuyển dữ liệu qua các ranh giới vật lý như: AppDomain, quy trình xử lý, và các ranh giới giao diện dịch vụ phải tuần tự hóa dữ liệu.
- Cũng có thể tuần tự hóa dữ liệu khi vượt qua các ranh giới logic (tuy nhiên, cần xem tác động hiệu suất).

# Bước 3: Xác định hỗ trợ tuần tự hóa

- **Hiển thị các thực thể nghiệp vụ tuần tự hóa trực tiếp chỉ khi được yêu cầu.**
  - Nếu lớp ứng dụng khác trong ứng dụng (trên cùng tầng vật lý) đang tiêu thụ các thực thể nghiệp vụ, cách đơn giản nhất là hiển thị trực tiếp các thực thể nghiệp vụ thông qua tuần tự hóa.
  - Nhược điểm hướng tiếp cận này là tạo ra phụ thuộc giữa người tiêu dùng các thực thể nghiệp vụ và triển khai của chúng.

# Bước 3: Xác định hỗ trợ tuần tự hóa

- **Hiện thị các thực thể nghiệp vụ tuần tự hóa trực tiếp chỉ khi được yêu cầu.**
  - Do đó, cách tiếp cận này không được khuyến nghị trừ khi muốn duy trì quyền kiểm soát trực tiếp đối với người tiêu dùng các thực thể nghiệp vụ và truy cập từ xa đến các thực thể nghiệp vụ giữa các tầng vật lý không được yêu cầu.

# Bước 3: Xác định hỗ trợ tuần tự hóa

- Chuyển đổi các thực thể nghiệp vụ thành các đối tượng chuyển đổi dữ liệu tuần tự hóa.
  - Để tách người tiêu dùng dữ liệu từ triển khai bên trong lớp nghiệp vụ, xem xét chuyển các thực thể nghiệp vụ thành các đối tượng chuyển đổi dữ liệu tuần tự hóa đặc biệt.
  - Đối tượng chuyển đổi dữ liệu (DTO) là một mẫu thiết kế sử dụng đóng gói nhiều cấu trúc dữ liệu thành một cấu trúc duy nhất cho việc truyền qua các ranh giới.

# Bước 3: Xác định hỗ trợ tuần tự hóa

- Chuyển đổi các thực thể nghiệp vụ thành các đối tượng chuyển đổi dữ liệu tuần tự hóa.
  - Các đối tượng chuyển đổi dữ liệu cũng rất hữu ích khi những người tiêu dùng các thực thể nghiệp vụ có một mô hình hay trình bày dữ liệu khác nhau.
  - Hướng tiếp cận này làm thay đổi việc triển khai bên trong lớp nghiệp vụ mà không ảnh hưởng bất kỳ các người tiêu thụ dữ liệu, và cho phép phiên bản các giao diện dễ dàng hơn.

# Bước 3: Xác định hỗ trợ tuần tự hóa

## ➤ Hiện thị trực tiếp XML.

- Một số trường hợp, có thể tuần tự hóa và hiển thị các thực thể nghiệp vụ dưới dạng XML.
- .NET Framework cung cấp hỗ trợ tuần tự hóa rộng rãi cho dữ liệu XML.
- Hầu hết các trường hợp, các thuộc tính trên các thực thể nghiệp vụ được sử dụng kiểm soát tuần tự hóa thành XML.



