

# Chương 5:

## Thiết kế dữ liệu lưu trữ



**GVLT:**

**TS. Trần Minh Triết – ThS. Đặng Bình Phương**  
[tmtriet@fit.hcmus.edu.vn](mailto:tmtriet@fit.hcmus.edu.vn) [dbphuong@fit.hcmus.edu.vn](mailto:dbphuong@fit.hcmus.edu.vn)

- ❖ Lưu trữ dữ liệu bằng CSDL quan hệ
  - Một số kỹ thuật để ánh xạ sơ đồ lớp sang sơ đồ logic (tổ chức lưu trữ dữ liệu bằng CSDL quan hệ)
- ❖ Lưu trữ dữ liệu bằng XML
  - Giới thiệu sơ lược về XML
  - Cách lưu trữ dữ liệu
- ❖ So sánh giữa cách sử dụng lưu trữ bằng XML, lưu trữ bằng CSDL quan hệ, và kết hợp cả 2 cách.

# Lưu trữ dữ liệu bằng CSDL quan hệ



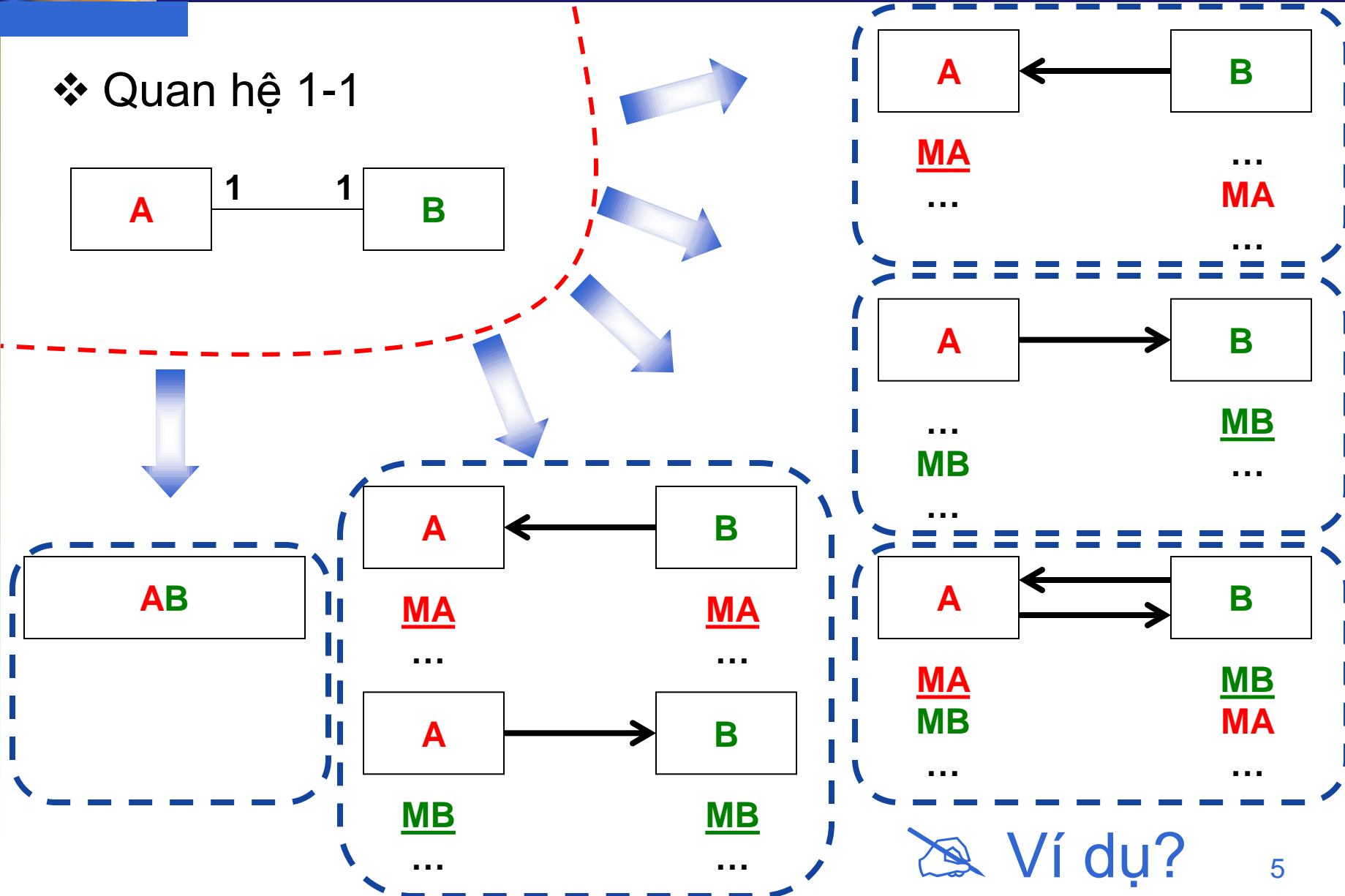
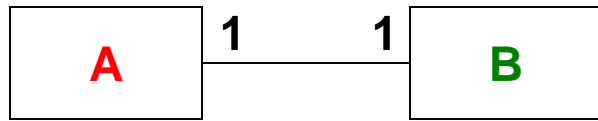
# Quy tắc #1

- ❖ Thông thường, mỗi lớp đối tượng đơn giản được ánh xạ thành **một bảng**

 Ví dụ?

# Quy tắc #2

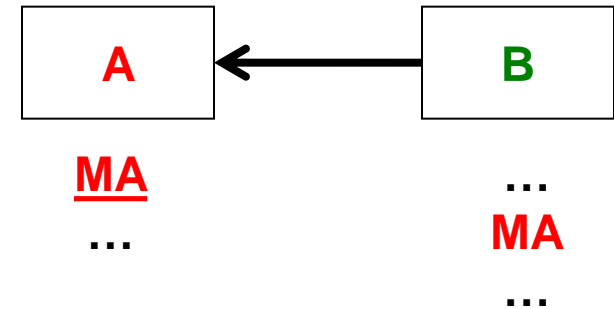
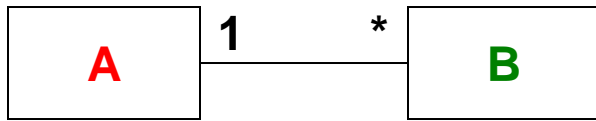
❖ Quan hệ 1-1



 Ví dụ?

# Quy tắc #3

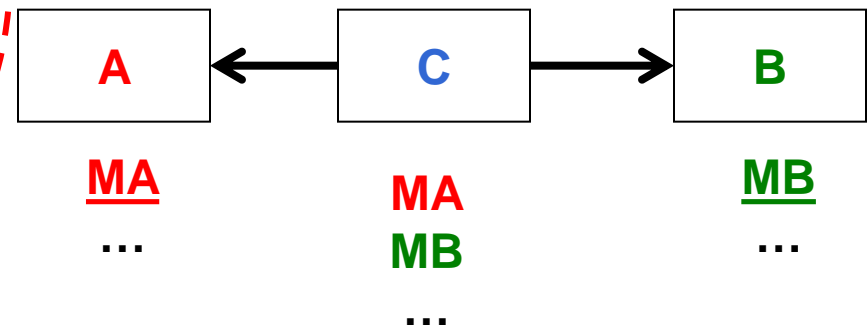
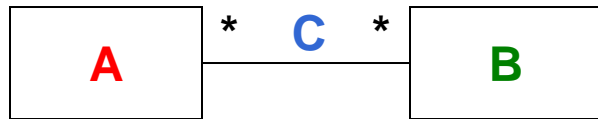
## ❖ Quan hệ 1-n



 Ví dụ?

# Quy tắc #4

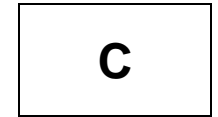
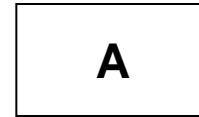
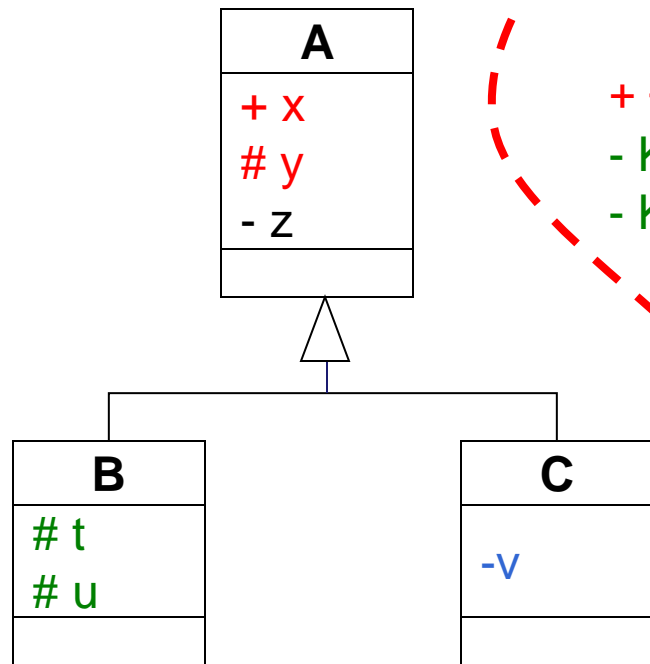
## ❖ Quan hệ m-n



 Ví dụ?

# Quy tắc #5

## ❖ Quan hệ kế thừa



$A(\underline{MA}, x, y, z)$     $B(\underline{MB}, x, y, t, u)$     $C(\underline{MC}, x, y, v)$

+ Đơn giản

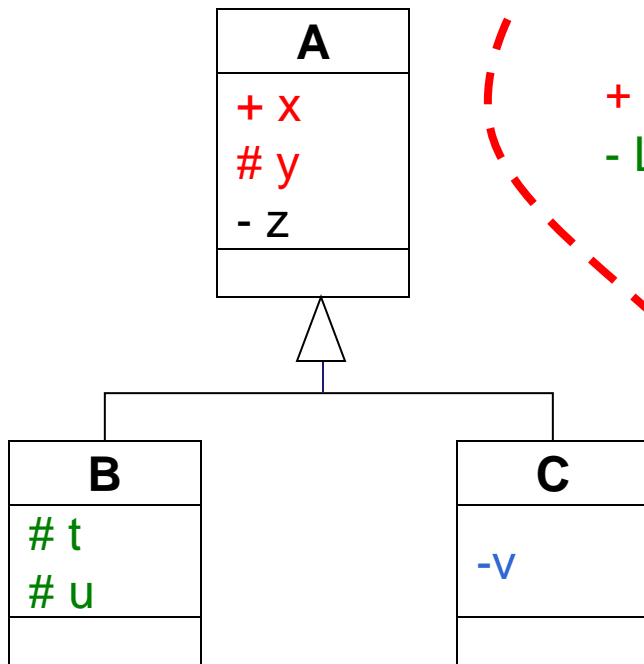
- Không thấy mối liên hệ giữa các loại đối tượng A, B, C
- Khó thống kê tổng quát

 Ví dụ?

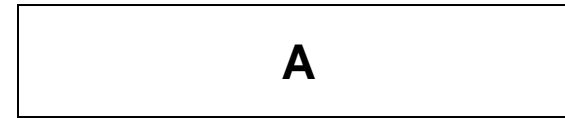


# Quy tắc #5

## ❖ Quan hệ kế thừa



- + Có được cái nhìn tổng quát về các đối tượng
- Lãng phí không gian lưu trữ



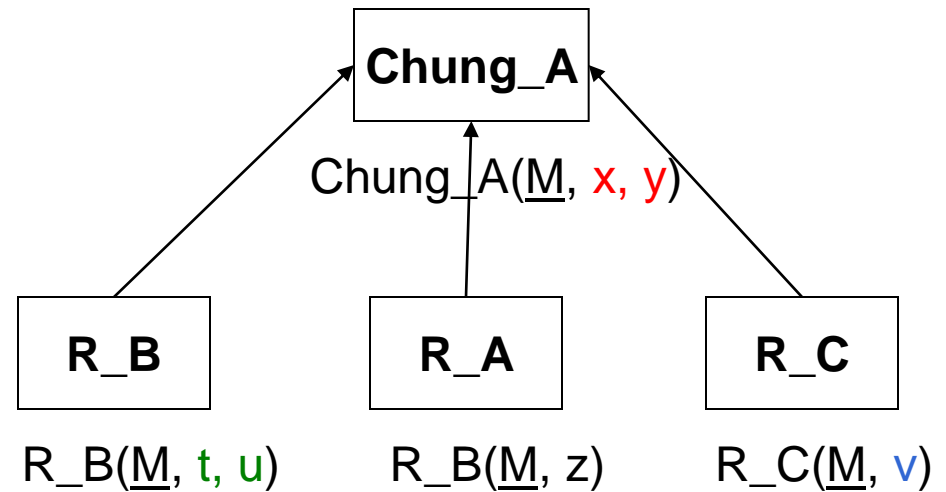
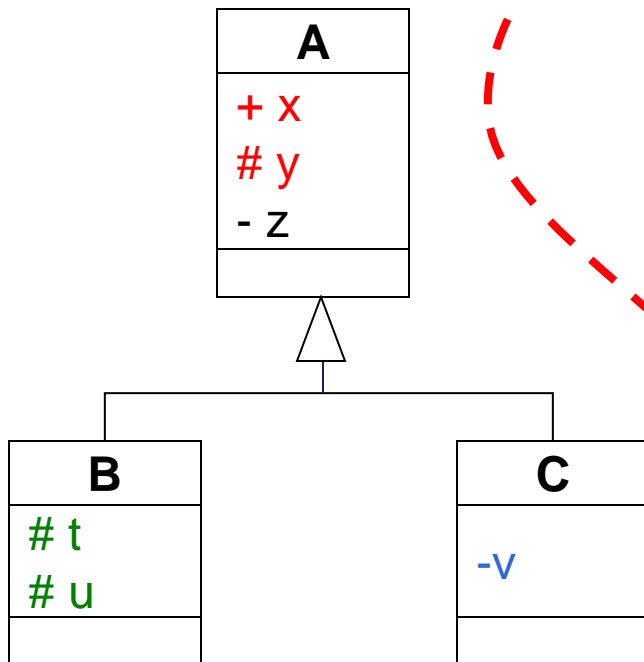
$A(\underline{M}, \text{Loại}, x, y, z, t, u, v)$

Loại	x	y	z	t	u	v
A						
B						
C						

 Ví dụ?

# Quy tắc #5

## ❖ Quan hệ kế thừa



- + Tiết kiệm không gian lưu trữ
- + Cho phép có cái nhìn tổng quát
- Tổ chức khá phức tạp

 Ví dụ?

## Quy tắc #6

- ❖ Lớp đối tượng có thuộc tính có **cấu trúc phức tạp**
- ❖ **Tách thành bảng phụ** để lưu trữ thuộc tính có cấu trúc phức tạp đó

class A

{

...

B x;

...

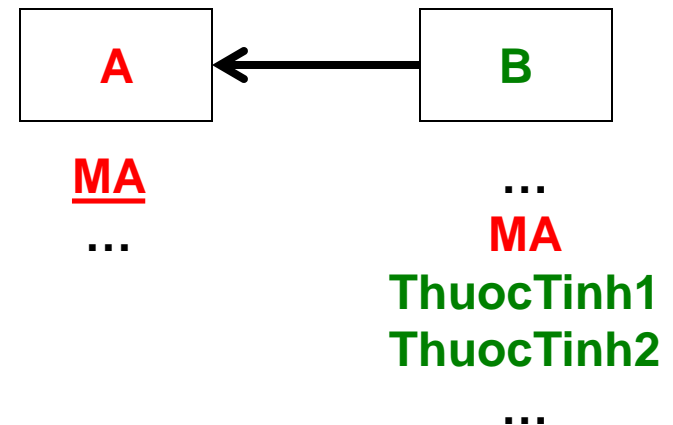
}

Cấu trúc của B gồm:

Thuộc tính 1

Thuộc tính 2

...

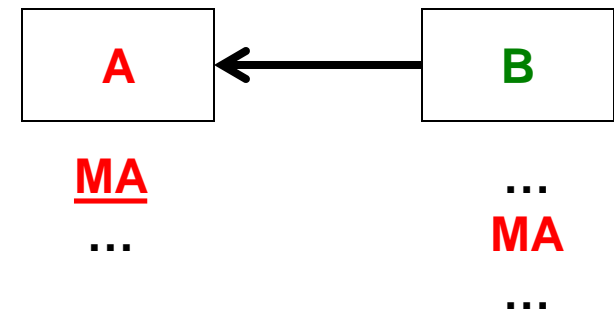


 Ví dụ?

# Quy tắc #7

- ❖ Lớp đối tượng có thuộc tính kiểu mảng
- ❖ Tách thành bảng chi tiết

```
class A
{
...
  B[] x;
...
}
```



 Ví dụ?

## Quy tắc #8

❖ Thuộc tính có giá trị rời rạc: Tách thành bảng danh mục

 Ví dụ?

# Quy tắc #9

## ❖ Bảng tham số

### ◦ Dạng 1

Khóa	ThamSố#1	ThamSố#2	...	ThamSố#N
...	...	...	...	...

Mỗi tham số tương ứng với một cột trong bảng tham số

Bảng tham số thường chỉ gồm 1 dòng (chứa giá trị các tham số hiện hành)



Cần bổ sung tham số mới?

Cần vô hiệu hóa tác dụng của một tham số?

# Quy tắc #9

## ❖ Bảng tham số

### • Dạng 2

MãThamSố	TênThamSố	Kiểu	GiáTrị	TìnhTrạng
...	...	...	...	...

Được lưu  
dạng chuỗi

Mỗi tham số tương ứng với một dòng trong bảng tham số

Giá trị hiện tại của tham số được lưu bằng dạng chuỗi

Mỗi tham số cần lưu trữ kiểu giá trị để phần mềm “hiểu” đúng nội dung giá trị hiện tại của tham số



Cần bổ sung tham số mới?

Cần vô hiệu hóa tác dụng của một tham số?

# Lưu trữ dữ liệu bằng XML

Lược dịch từ slide chương 2 – XML  
trong môn Managing XML and Semistructured Data  
của GS. Dan Suciu,  
University of Washington

Department of Computer Science & Engineering

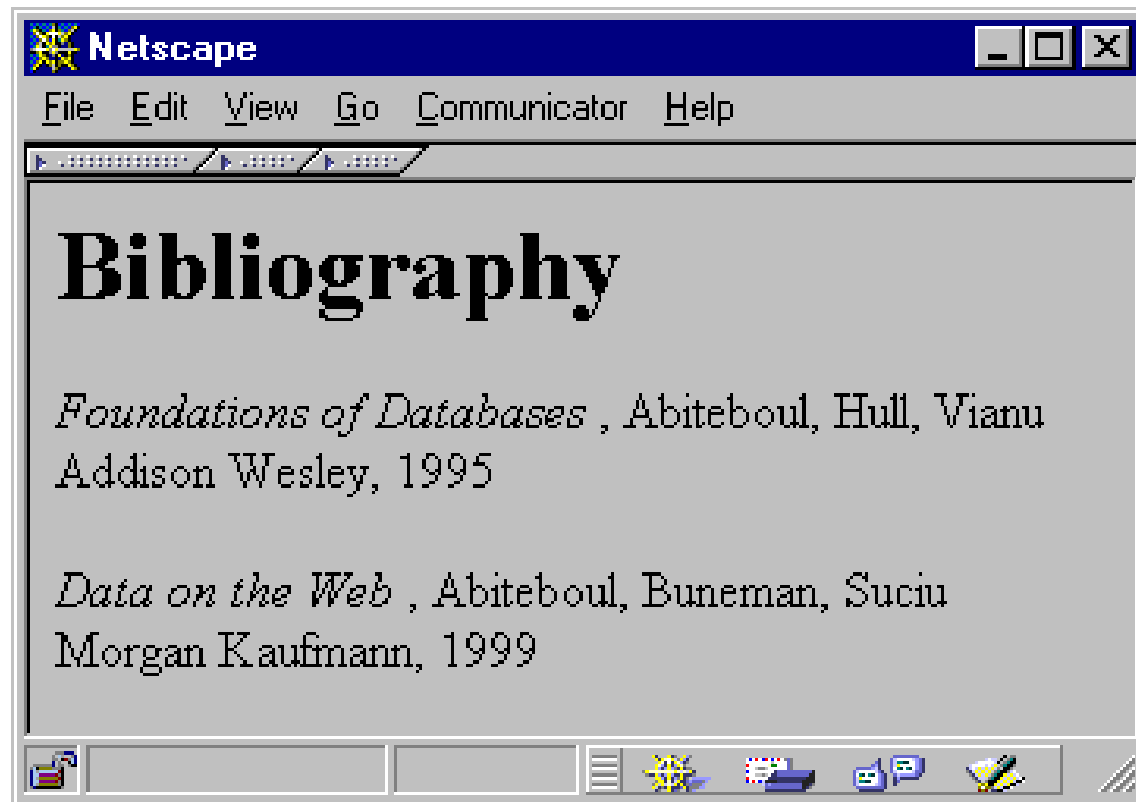
Tài liệu gốc:

<http://www.cs.washington.edu/homes/suciu/COURSES/590DS/02xmlsyntax.ppt>



- ❖ Chuẩn của W3C
- ❖ Ý tưởng ban đầu: văn bản có cấu trúc (SGML)
- ❖ <http://www.w3.org/TR/2000/REC-xml-20001006>  
(version 2, 10/2000)  $\text{HTML4.0} \in \text{XML} \subset \text{SGML}$

# HTML và XML



HTML mô tả “cách thể hiện”

`<h1> Bibliography </h1>`

`<p> <i> Foundations of Databases </i>`

Abiteboul, Hull, Vianu

`<br>` Addison Wesley, 1995

`<p> <i> Data on the Web </i>`

Abiteoul, Buneman, Suciu

`<br>` Morgan Kaufmann, 1999

```
<bibliography>
  <book>
    <title> Foundations... </title>
    <author> Abiteboul </author>
    <author> Hull </author>
    <author> Vianu </author>
    <publisher> Addison Wesley </publisher>
    <year> 1995 </year>
  </book>
  ...
</bibliography>
```

XML mô tả nội dung

<bibliography>

<book>

Start tag

<title> Foundations... </title>

<author> Abiteboul </author>

<author> Hull </author>

<author> Vianu </author>

<publisher> Addison Wesley </publisher>

<year> 1995 </year>

</book>

...

</bibliography>

End tag



# XML

```
<bibliography>
```

```
  <book>
```

```
    <title> Foundations... </title>
```

```
    <author> Abiteboul </author>
```

```
    <author> Hull </author>
```

```
    <author> Vianu </author>
```

```
    <publisher> Addison Wesley </publisher>
```

```
    <year> 1995 </year>
```

```
  </book>
```

```
  ...
```

```
</bibliography>
```

- ❖ tag: **book**, **title**, **author**, ...
- ❖ start tag: **<book>**, end tag: **</book>**
- ❖ element: **<book>...<book>**, **<author>...</author>**
- ❖ Các element có thể lồng nhau
- ❖ Element rỗng: **<red></red>** hay **<red/>**
- ❖ Một tài liệu XML: có 1 *root element*

Tài liệu XML được gọi là “**well formed**” nếu các **tag bắt đầu** và **tag kết thúc** tương ứng xuất hiện **đầy đủ** và **đúng vị trí** (kể cả với các tag lồng nhau)



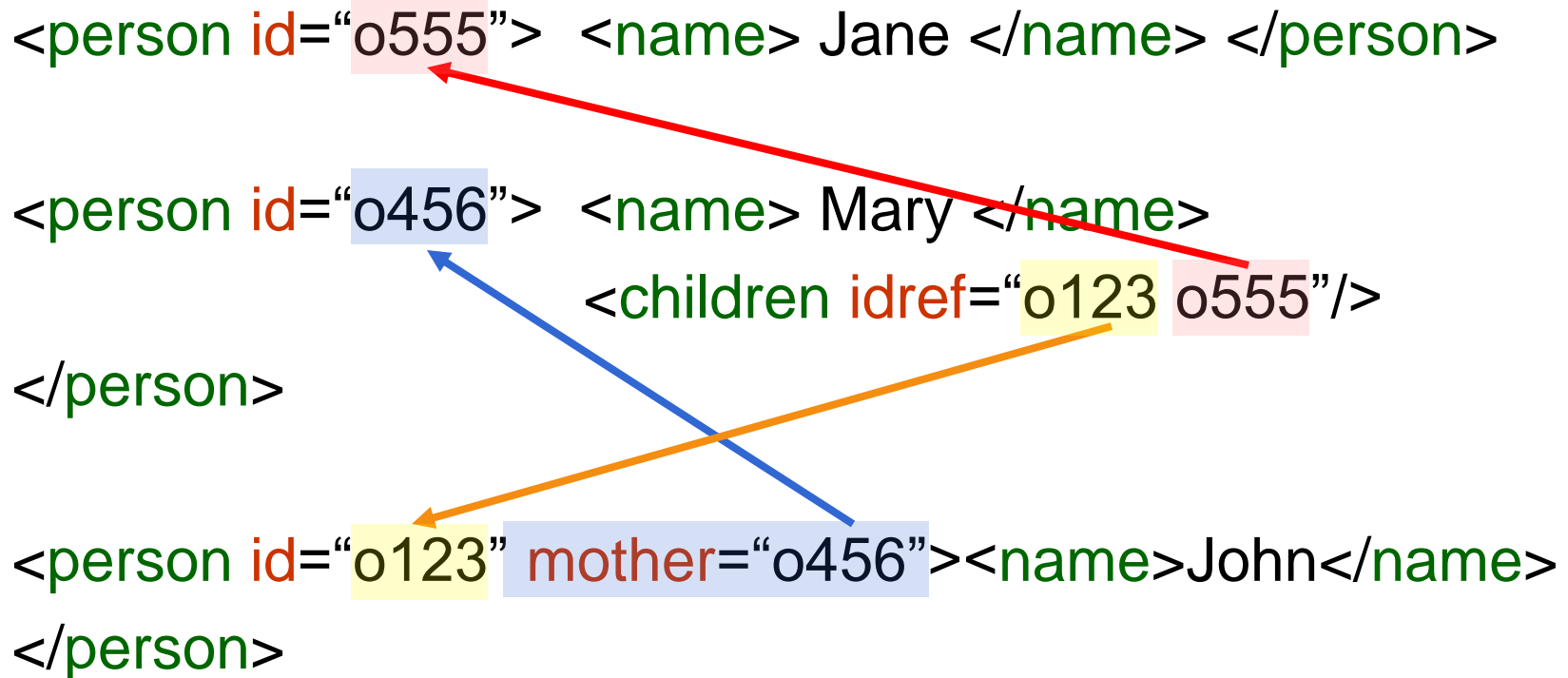
# XML: Attribute

```
<book price = "55" currency = "USD">  
  <title> Foundations of Databases </title>  
  <author> Abiteboul </author>  
  ...  
  <year> 1995 </year>  
</book>
```

Attribute: cách khác để biểu diễn dữ liệu  
(không dùng Element)



# XML: Oid và Tham chiếu



The diagram illustrates XML references using three code snippets. The first snippet shows a person with ID "o555". The second snippet shows a person with ID "o456" who has children, with one child having ID "o123" and another having ID "o555". The third snippet shows a person with ID "o123" whose mother is "o456". Arrows indicate the following references: a red arrow from "o555" in the first snippet to "o555" in the second snippet's children list; a blue arrow from "o456" in the second snippet to "o456" in the third snippet's mother attribute; and an orange arrow from "o123" in the second snippet's children list to "o123" in the third snippet's ID attribute.

```
<person id="o555"> <name> Jane </name> </person>  
  
<person id="o456"> <name> Mary </name>  
  <children idref="o123 o555"/>  
</person>  
  
<person id="o123" mother="o456"><name>John</name>  
</person>
```

OID và tham chiếu trong XML

# Tham chiếu đến entity

❖ Cú pháp: &entityname;

❖ Ví dụ:

<element> this is less than &lt; </element>

❖ Một số entity thông dụng:

&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	“
&#38;	Unicode char



# Sử dụng Comment

<!-- .... Comment text... -->



# XML Namespace

- ❖ <http://www.w3.org/TR/REC-xml-names> (1/99)
- ❖  $\text{name} ::= [\text{prefix:}] \text{localpart}$

```
<book xmlns:isbn="www.isbn-org.org/def">  
  <title> ... </title>  
  <number> 15 </number>  
  <isbn:number> .... </isbn:number>  
</book>
```

# XML Namespace

❖ Cú pháp: `<number>` , `<isbn:number>`

```
<tag xmlns:mystyle = "http://...">
```

...

```
<mystyle:title> ... </mystyle:title>
```

```
<mystyle:number> ...
```

```
</tag>
```

Định nghĩa tại...



# Mô hình truy vấn dữ liệu XML

❖  $\text{Node} ::= \text{DocNode} \mid$   
 $\text{ElemNode} \mid$   
 $\text{ValueNode} \mid$   
 $\text{AttrNode} \mid$   
 $\text{NSNode} \mid$   
 $\text{PINode} \mid$   
 $\text{CommentNode} \mid$   
 $\text{InfoltemNode} \mid$   
 $\text{RefNode}$



# Mô hình truy vấn dữ liệu XML

Nút Element:

❖ `elemNode` : (`QNameValue`,  
                  `{AttrNode }`,  
                  `[ ElemNode | ValueNode ]`)  
          → `ElemNode`

- ❖ `QNameValue` = “tên tag”
- ❖ `{...}` = “tập hợp gồm ...”
- ❖ `[...]` = “danh sách gồm ...”

# Mô hình truy vấn dữ liệu XML

## Ví dụ

```
<book price = "55"  
      currency = "USD">  
  <title> Foundations ... </title>  
  <author> Abiteboul </author>  
  <author> Hull </author>  
  <author> Vianu </author>  
  <year> 1995 </year>  
</book>
```

```
book1= elemNode(book,  
  {price2, currency3},  
  [title4,  
    author5,  
    author6,  
    author7,  
    year8])
```

```
price2 = attrNode(...) /* next */  
currency3 = attrNode(...)  
title4 = elemNode(title, string9)  
...
```





# Mô hình truy vấn dữ liệu XML

Nút Attribute:

❖ attrNode : (QNameValue, ValueNode)  
→ AttrNode



# Mô hình truy vấn dữ liệu XML

## Ví dụ

```
<book price = "55"  
      currency = "USD">  
  <title> Foundations ... </title>  
  <author> Abiteboul </author>  
  <author> Hull </author>  
  <author> Vianu </author>  
  <year> 1995 </year>  
</book>
```

```
price2 = attrNode(price,string10)  
string10 = valueNode(...) /* next */  
currency3 = attrNode(currency,  
                      string11)  
string11 = valueNode(...)
```



# Mô hình truy vấn dữ liệu XML

Nút Value:

- ❖  $\text{ValueNode} = \text{StringValue} \mid \text{BoolValue} \mid \text{FloatValue} \dots$
- ❖  $\text{stringValue} : \text{string} \rightarrow \text{StringValue}$
- ❖  $\text{boolValue} : \text{boolean} \rightarrow \text{BoolValue}$
- ❖  $\text{floatValue} : \text{float} \rightarrow \text{FloatValue}$



# Mô hình truy vấn dữ liệu XML

## Ví dụ

```
<book price = "55"  
      currency = "USD">  
  <title> Foundations ... </title>  
  <author> Abiteboul </author>  
  <author> Hull </author>  
  <author> Vianu </author>  
  <year> 1995 </year>  
</book>
```

```
price2 = attrNode(price,string10)  
string10 = valueNode(stringValue("55"))  
currency3 = attrNode(currency, string11)  
string11 = valueNode(stringValue("USD"))  
title4 = elemNode(title, string9)  
string9 =  
valueNode(stringValue("Foundations..."))
```



# Luyện tập

- ❖ Làm quen việc sử dụng XML để lưu trữ thông tin (thay cho cơ sở dữ liệu)
- ❖ Mở rộng:
  - Tìm hiểu cách khai báo XML Schema
  - Tìm hiểu về XQuery

# So sánh giữa cách sử dụng lưu trữ bằng CSDL và bằng XML





# So sánh

## Sử dụng CSDL

- ❖ Ưu điểm
- ❖ Nhược điểm
- ❖ Trường hợp nên sử dụng

## Sử dụng XML

- ❖ Ưu điểm
- ❖ Nhược điểm
- ❖ Trường hợp nên sử dụng

**Thảo luận  
trực tiếp  
tại lớp**