

IIC3432 - Tópicos Avanzados en Bases de Datos

Una introducción a XML

Documentos versus Bases de Datos

Documentos	Base de Datos
estáticos estructura implícita semi-estructurados fácil de entender para una persona importa: presentación, edición, codificación de caracteres, lenguaje	dinámicos estructura explícita estructurados fácil de entender para un computador importa: modelo, consultas, eficiencia

Un poco de historia: SGML, HTML y XML

- SGML: Standard Generalized Markup Language.
 - Charles Goldfarb, 1986.
 - DTD (Document Type Definition).
 - Herramienta poderosa y flexible para estructurar información.
 - Implementaciones resultaron ser demasiado complicadas.
 - Herramientas para trabajar con SGML no eran eficientes.
- Dos sub-lenguajes:
 - HTML: HyperText Markup Language (Tim Berners-Lee, 1991).
 - **XML**: e**X**tensible **M**arkup **L**anguage, W3C, 1998.

¿Por qué estudiar XML?

- Hay una enorme demanda por intercambio de información.
 - Entre plataformas.
 - Entre empresas.
- Hay una enorme demanda por integración de datos.
 - Fuentes de datos heterogéneos.
 - Fuentes de datos distribuidas en lugares distintos.
- **XML:** Lenguaje estándar para intercambiar información en la Web y un modelo de datos muy usado en la integración de bases de datos.

¿Por qué no usar HTML como base de datos?

```
<UL>
  <LI> Book:
    <UL>
      <LI> Algebra </LI>
      <LI> Hungerford </LI>
    </UL>
  </LI>
  <LI> Book:
    <UL>
      <LI> Real Analysis </LI>
      <LI> Royden </LI>
    </UL>
  </LI>
</UL>
```

¿Por qué no usar HTML como base de datos?

- Un cambio en el formato puede hacer que demos una respuesta incorrecta a una pregunta.
- HTML tags:
 - Predefinidos y fijos.
 - Describen el formato de salida en lugar de la estructura de la información.
- HTML es bueno para presentar información, pero no ayuda en la extracción de información.

XML versus HTML

- XML tags:
 - Definidos por el usuario.
 - Describen la estructura de la información.
- XML es un buen formato tanto para las personas como para el computador.
 - HTML sólo es bueno para las personas.

XML: Un primer ejemplo

```
<db>
  <book>
    <title> Algebra </title>
    <author>
      <name> Hungerford </name>
    </author>
  </book>
  <book>
    <title> Real Analysis </title>
    <author>
      <name> Royden </name>
    </author>
  </book>
</db>
```

XML: Estructura

- Nuevos tags definidos por el usuario: `<book>`, `<title>`, ...
- Estructura puede estar anidada arbitrariamente:

```
<db>
```

```
  <book>
```

```
    <title> Algebra </title>
```

```
    <author>
```

```
      <name> Hungerford </name>
```

```
    </author>
```

```
  ...
```

XML: Estructura

- Estructura puede estar definida recursivamente:

```
<db>
```

```
  <book>
```

```
    <section>
```

```
      <title> Automata Theory </title>
```

```
      <number> 2 </number>
```

```
      <section>
```

```
        <title> String automata </title>
```

```
        <number> 2.1 </number>
```

```
      </section>
```

```
    </section>
```

```
  ...
```

XML: Presentación

- XML no define como debe desplegarse la información almacenada.
- Style sheet: Contiene reglas que indican como desplegar el documento.
 - Puede indicar como transformar XML a HTML.
 - XSL (eXtensible Style Language).

Tags

- Un documento XML contiene **tags** y texto:

```
<author>  
  <name> Hungerford </name>  
</author>
```

- Tags vienen en pares:
 - tag **inicial:** **<author>**
 - tag **final:** **</author>**

Tags

- Tags tienen que estar anidados correctamente:
 - Correcto: `<author> <name> ... </name> </author>`
 - Incorrecto: `<author> <name> ... </author> </name>`
- XML tiene sólo un tipo básico: PCDATA (Parsed Character DATA).

Elementos en XML

- Elemento: Segmento entre un tag inicial y su correspondiente tag final.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

Elementos en XML

- Elemento: Segmento entre un tag inicial y su correspondiente tag final.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

Elementos en XML

- Sub-elemento: Relación entre un elemento y los elementos que lo componen.

`<person>`

`<name> Leonid Libkin </name>`

`<email> libkin@cs.toronto.edu </email>`

`<oldemail> libkin@cs.nyu.edu </oldemail>`

`<oldemail> libkin@research.bell-labs.com </oldemail>`

`<oldemail> leonid.libkin@inria.fr </oldemail>`

`</person>`

Elementos en XML

- Sub-elemento: Relación entre un elemento y los elementos que lo componen.

`<person>`

`<name> Leonid Libkin </name>`

`<email> libkin@cs.toronto.edu </email>`

`<oldemail> libkin@cs.nyu.edu </oldemail>`

`<oldemail> libkin@research.bell-labs.com </oldemail>`

`<oldemail> leonid.libkin@inria.fr </oldemail>`

`</person>`

Elementos en XML

- Están ordenados.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

Elementos en XML

- Están ordenados.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail> ⇐ 1
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

Elementos en XML

- Están ordenados.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail> ⇐ 2
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

Elementos en XML

- Están ordenados.

```
<person>
  <name> Leonid Libkin </name>
  <email> libkin@cs.toronto.edu </email>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail> ⇐ 3
</person>
```

Elementos en XML

- Están ordenados: Los siguientes documentos son **considerados distintos**.

```
<person>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
</person>
```

```
<person>
  <oldemail> libkin@cs.nyu.edu </oldemail>
  <oldemail> leonid.libkin@inria.fr </oldemail>
  <oldemail> libkin@research.bell-labs.com </oldemail>
</person>
```

Elementos destacados

- Un documento XML está definido como un sólo elemento llamado **elemento raíz**.

`<db>`

`...`

`</db>`

- Elementos **vacíos**: Indican que no hay contenido.
 - `<author></author>` puede ser reemplazado por `<author/>`.

Atributos

- Un tag inicial puede contener **atributos** que describen propiedades del elemento:

```
<picture>  
  <height dim="cm"> 2400 </height>  
  <width dim="in"> 96 </width>  
  <name> foto1 </name>  
</picture>
```

Atributos: Estructura

- No pueden aparecer anidados.
- Pueden aparecer una sola vez en cada tag inicial.
 - `<height dim="cm" dim="mm">` es incorrecto.
- No están ordenados:
 - Son considerados iguales:
`<book title="Algebra" author="Hungerford">` y
`<book author="Hungerford" title="Algebra">`.

Documentos XML bien contruidos

Un documento XML está bien construido si:

- Tags están anidados de manera correcta.
- Cada atributo aparece a lo más una vez en cada tag inicial.

Estas condiciones son débiles: No dicen mucho sobre la estructura de los documentos.

Un paréntesis: Bases de datos relacionales y XML

Una base de datos relacional:

Sigla	Sección	Sala	Profesor
IIC1102	3	A8	Rosa Alarcón
IIC1102	4	A5	Marcelo Arenas
IIC3432	1	JP	Marcelo Arenas

¿Cómo se puede almacenar esta información en XML? ¿Cómo lo haría si tuviera varias relaciones?

Otros constructores

- Declaración XML: Información sobre la versión usada debe ser indicada.

```
<?xml version='1.0'?>
```

- Comentarios: Son ignorados por los procesadores.

```
<!-- Este es un comentario -->
```

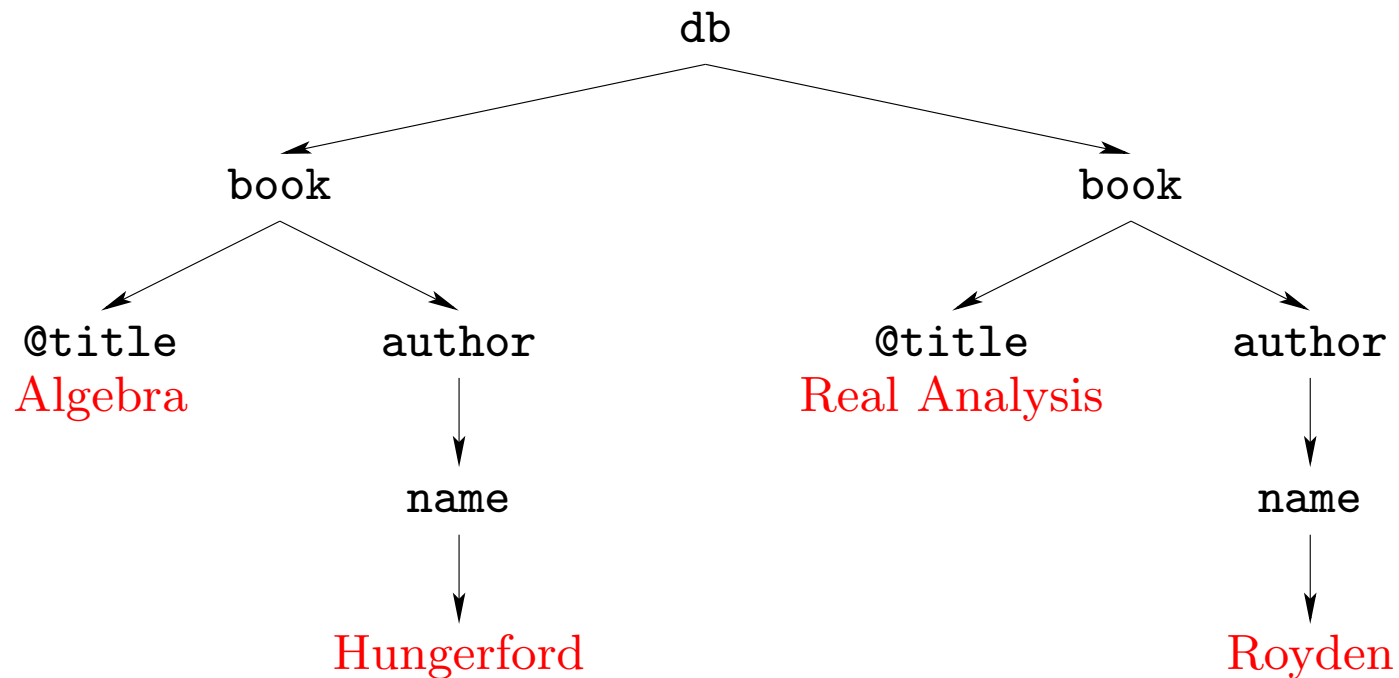
Un documento XML completo

```
<?xml version='1.0'?>
<db>
  <book>
    <title> Algebra </title>
    <author>
      <name> Hungerford </name>
    </author>
  </book>
  <book>
    <title> Real Analysis </title>
    <author>
      <name> Royden </name>
    </author>
  </book>
</db>
```

Otro documento XML completo

```
<?xml version='1.0'?>
<db>
  <book title="Algebra">
    <author>
      <name> Hungerford </name>
    </author>
  </book>
  <book title="Real Analysis">
    <author>
      <name> Royden </name>
    </author>
  </book>
</db>
```

Documentos XML como árboles



Para estudiar XML: Debemos saber un poco más sobre árboles ...