

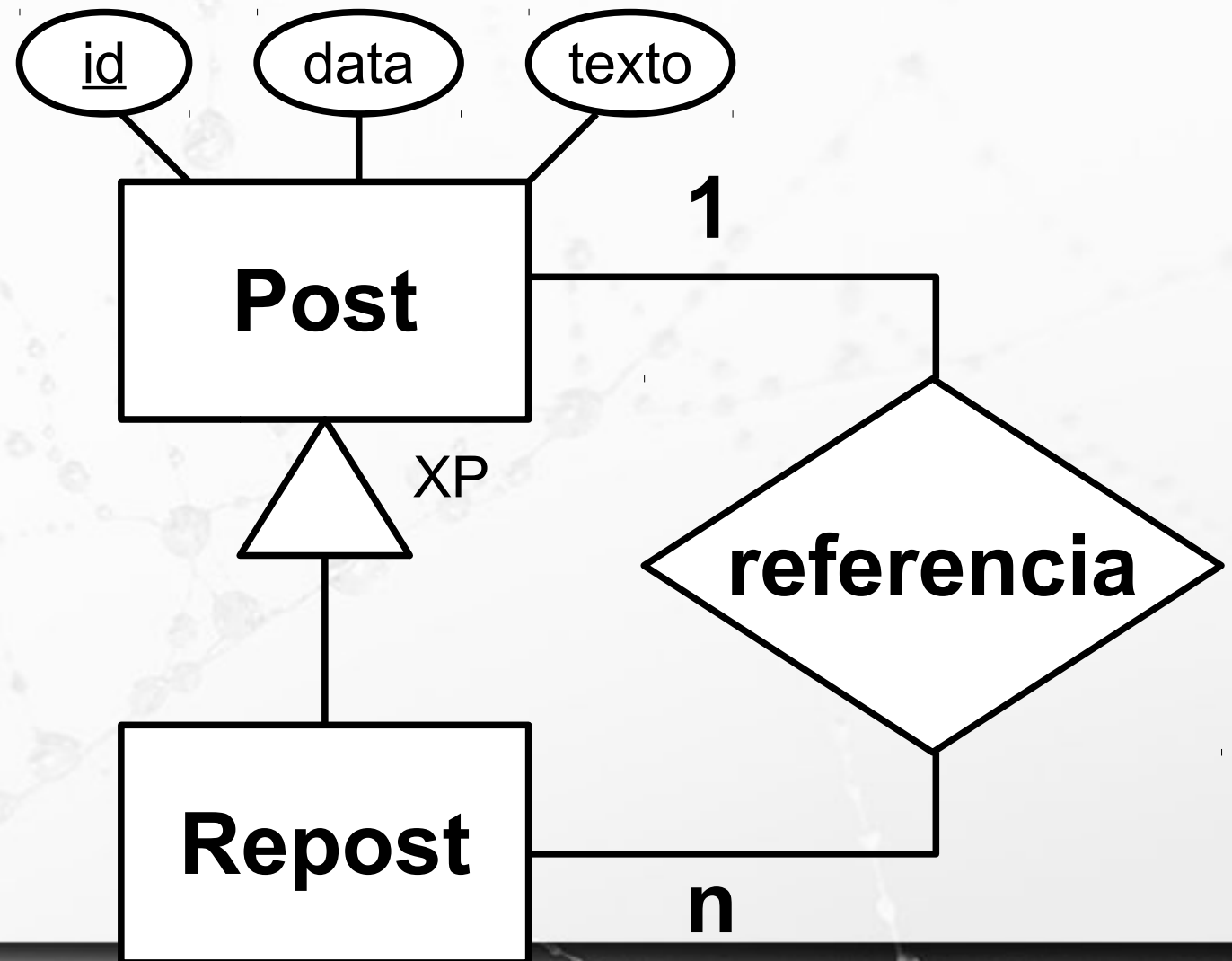
Prática: Relacional & XML

Banco de Dados: Teoria e Prática

André Santanchè e Patrícia Cavoto
Institute of Computing - UNICAMP
Novembro 2016

Exercício 1

- Elabore os possíveis esquemas relacionais que atendam a este modelo conceitual.



Exercício 1

- Solução 1 - Duas tabelas

Post(id, data, texto)

Repost(id, data, texto, ref_post)

→ ref_post - chave estrangeira Post(id)

Exercício 1

- Solução 2 - Uma tabela

Post(id, data, texto, ref_post)

→ ref_post - chave estrangeira Post(id)

→ ref_post=null → **post original**

ref_post<>null → **repost**

Exercício 2

- Dada a seguinte tabela Post, escreva uma sentença SQL que indique quantos **reposts** diretos ou indiretos tem cada **post original**. Um **repost indireto** é quando há uma cadeia de referências que vai do repost ao post original.

Post(id, data, texto, ref_post)

→ ref_post - chave estrangeira Post(id)

→ ref_post=null → **post original**

ref_post<>null → **repost**

Exercício 2

- Dada a seguinte tabela Post, escreva uma sentença SQL que indique quantos **reposts** diretos ou indiretos tem cada **post original**.
- Seria necessária recursividade no SQL. Não é possível com os operadores tratados em sala.

Post (id, data, texto, ref_post)

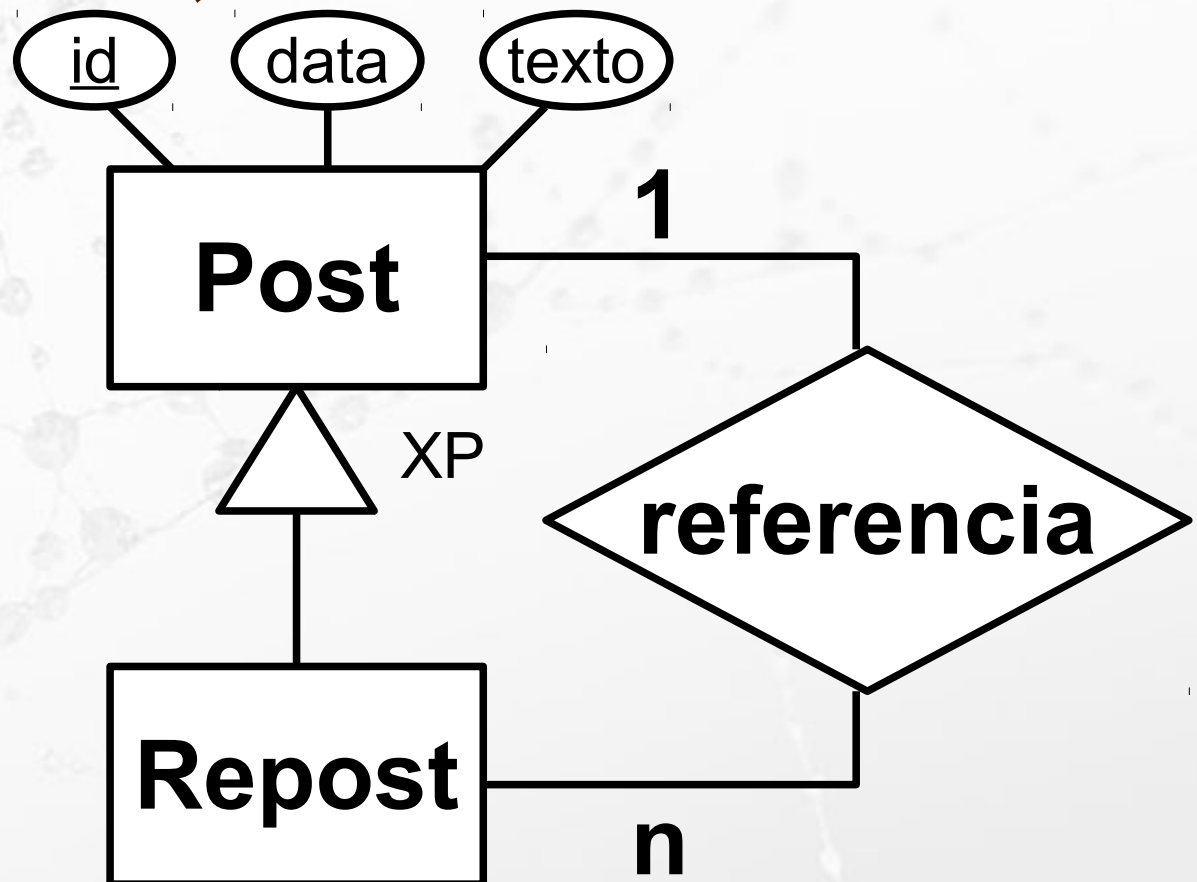
→ ref_post - chave estrangeira Post(id)

→ ref_post=null → **post original**

ref_post<>null → **repost**

Exercício 3

- Escreva exemplos de possíveis arquivos XML que sejam capazes de representar o modelo conceitual (um para cada abordagem de relacionamento - i.e., embedded e referência cruzada).

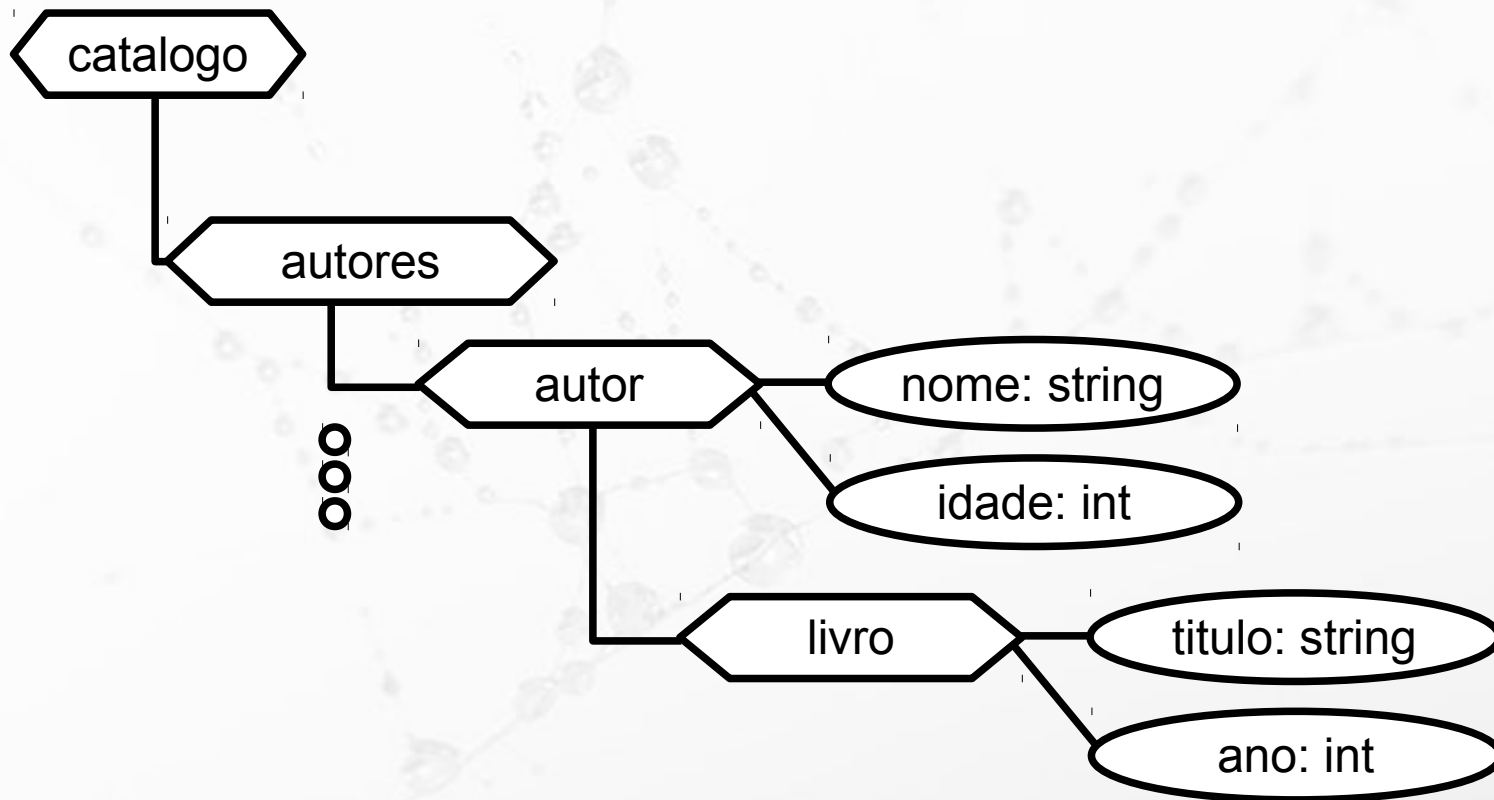


Modelos de Relações



Embedded

Autor/Livro Embedded



Livro Embedded em Autor

```
{  
  "nome": "Doriana",  
  "livro": {  
    "titulo": "Horacio o dino",  
    "ano": 2015  
  }  
}
```

563c871c45400d49a72753c4

nome: "Doriana"

livro

titulo: "Horacio o dino"
ano: 2015

Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Doriana">
```

```
    <livro titulo="Horacio o dino"  
ano="2015" />
```

```
  </autor>
```

```
  ...
```

```
</autores>
```

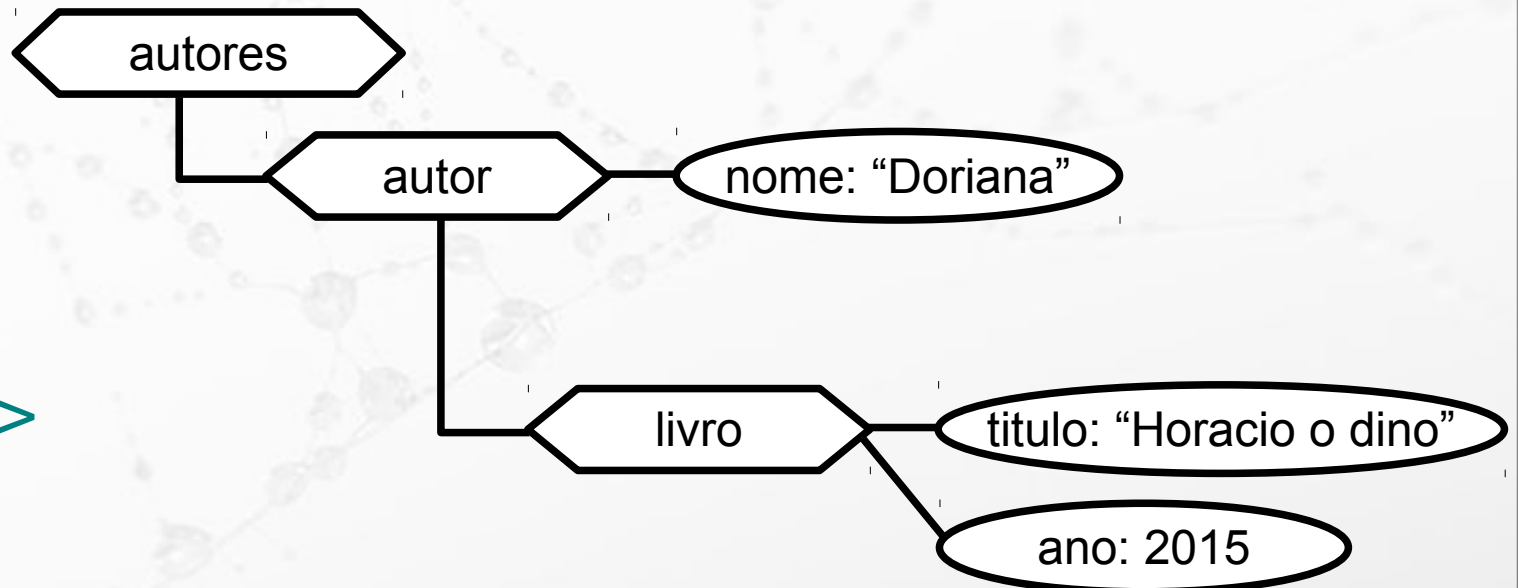
```
...
```

```
</catalogo>
```

563c871c45400d49a72753c4
nome: "Doriana"



titulo: "Horacio o dino" ano: 2015



Livro Embedded em Autor

```
{ "nome": "Asdrubal",  
  "idade": 25,  
  "livro": {  
    "titulo": "Vida sentimental dos  
dinossauros",  
    "ano": 2013  
  }  
}
```

563c873745400d49a72753c5

nome: "Asdrubal"
idade: 25

livro

titulo: "Vida sentimental dos dinossauros"
ano: 2013

Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Asdrubal" idade="25"
```

```
    <livro titulo="Vida sentimental dos dinossauros"  
ano="2013" />
```

```
  </autor>
```

```
...
```

```
</autores>
```

```
...
```

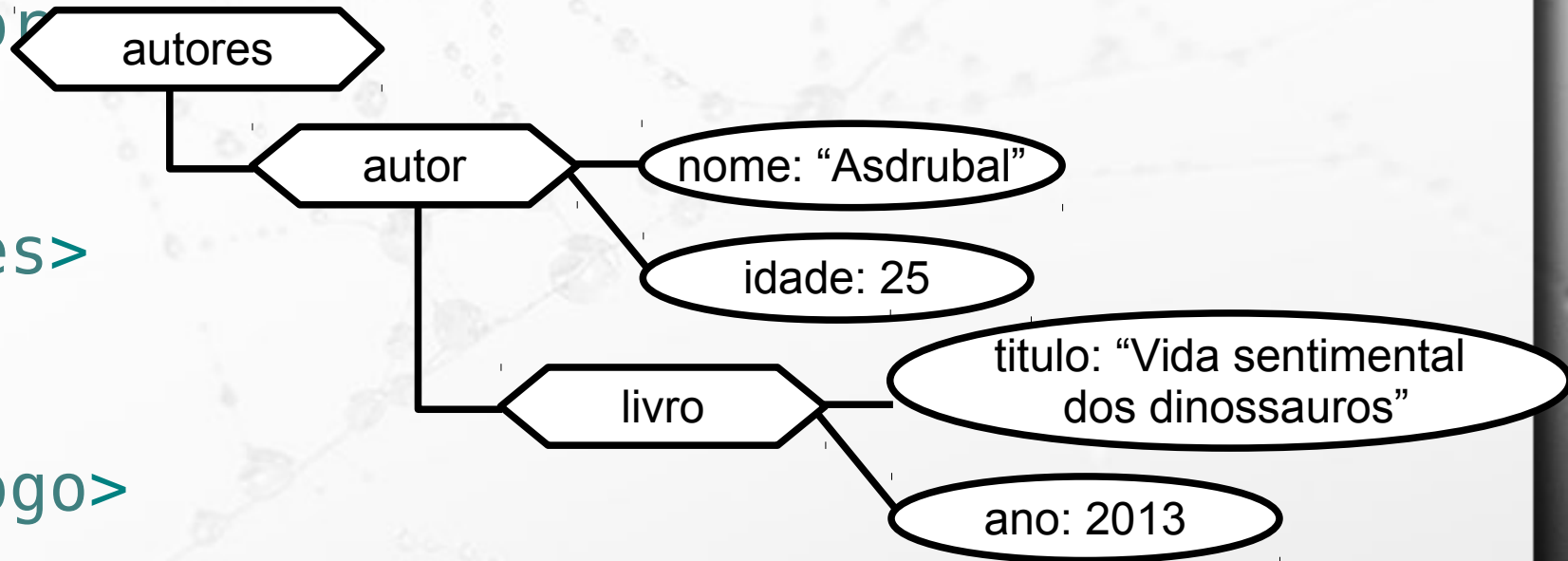
```
</catalogo>
```

563c873745400d49a72753c5

nome: "Asdrubal"
idade: 25

livro

titulo: "Vida sentimental dos dinossauros"
ano: 2013



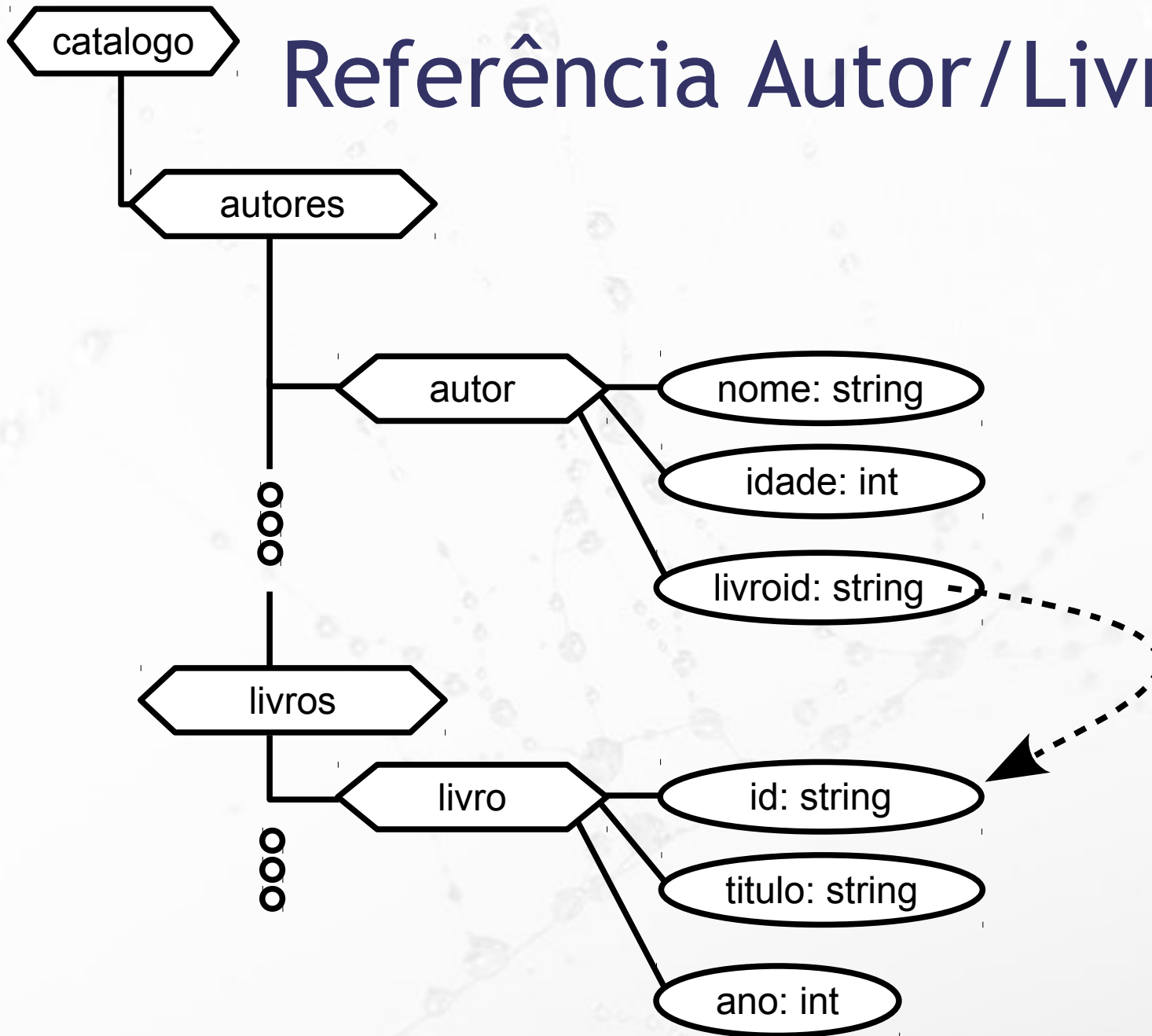
Exercício 3 - Embedded

<http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml>

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  <POST id="1010" data="2016-10-25">
    <TEXT0>Por que nao?</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
</POST>
</POST>
</POST>
<POST id="1317" data="2016-10-10">
  <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
<POST id="2021" data="2016-10-15">
  <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
<POST id="1818" data="2016-10-18">
  <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
</POST>
</POST>
</POST>
</FEED>
```

Referências Cruzadas

Referência Autor/Livro



Insert com “_id” manual bloco livros

```
{  
  "_id": "livro-como-entender",  
  "titulo": "Como entender seu dino",  
  "ano": 2015  
}
```

livro-como-entender
titulo: "Como entender seu dino" ano: 2015

Referência Autor/Livro

```
<catalogo>
```

```
<livros>
```

```
<livro id="livro-como-entender"  
      titulo="Como entender seu dino"  
      ano="2015" />
```

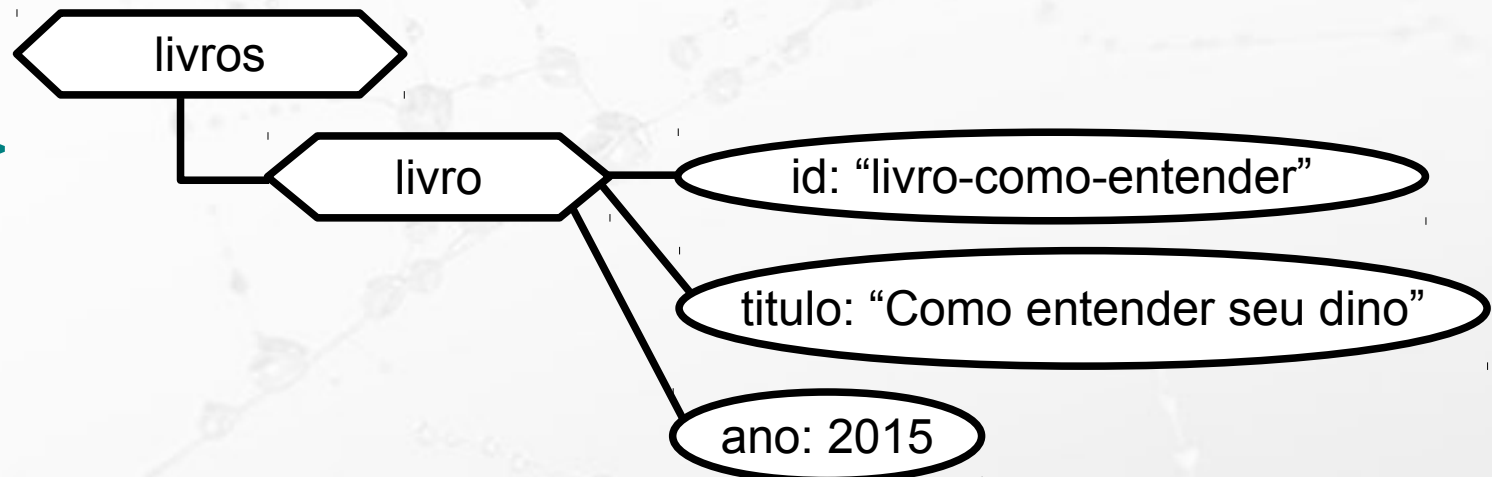
```
...
```

```
</livros>
```

```
...
```

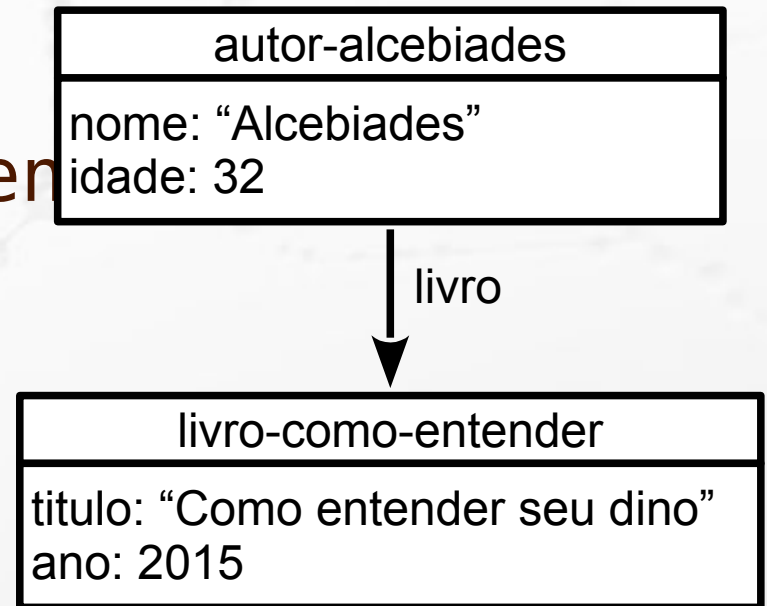
```
</livro>
```

livro-como-entender
titulo: "Como entender seu dino"
ano: 2015

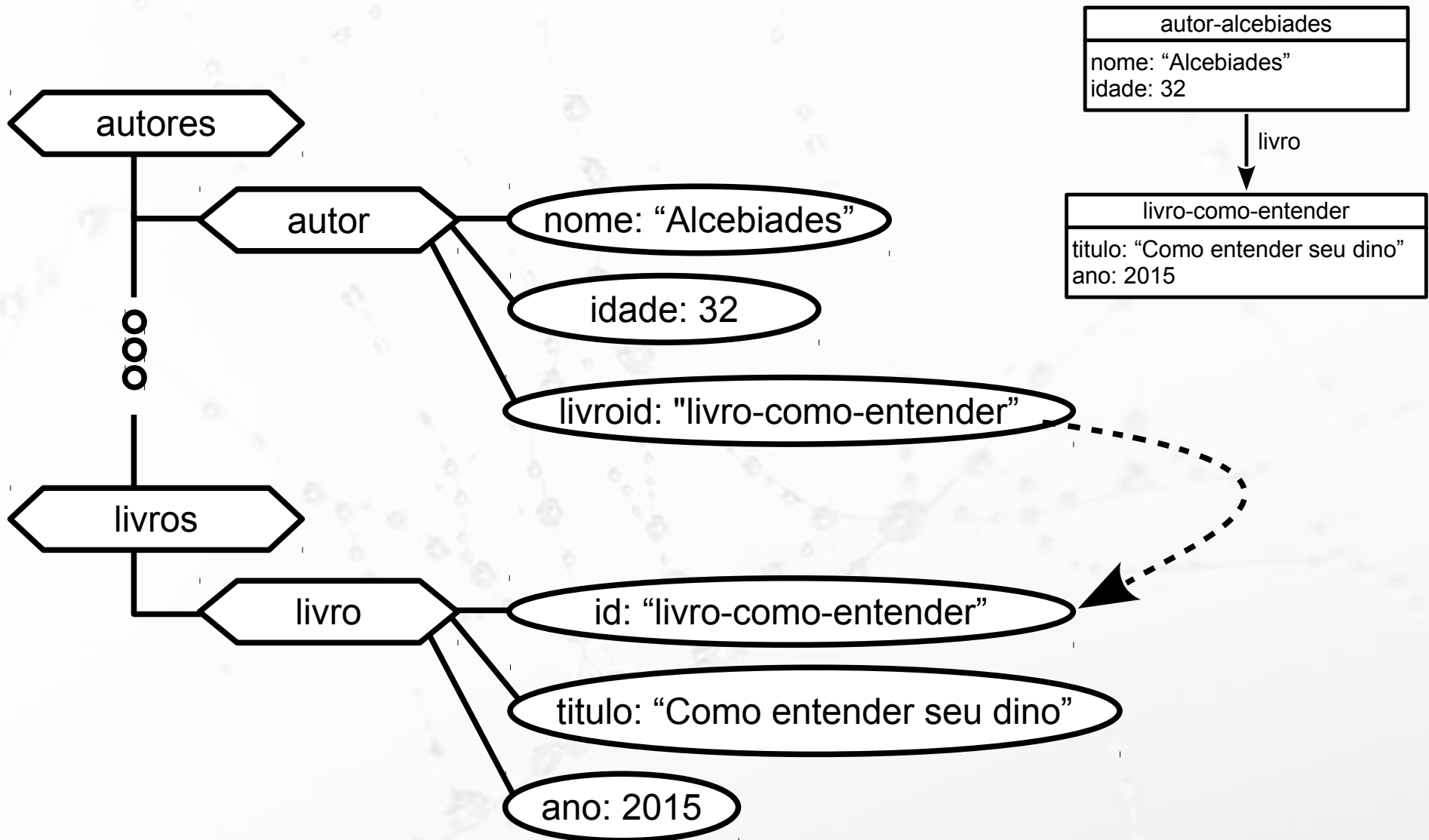


Insert referência a livro bloco autores

```
{  
  "_id": "autor-alcebiades",  
  "nome": "Alcebiades",  
  "idade": 32,  
  "livro": "livro-como-entender"  
}
```

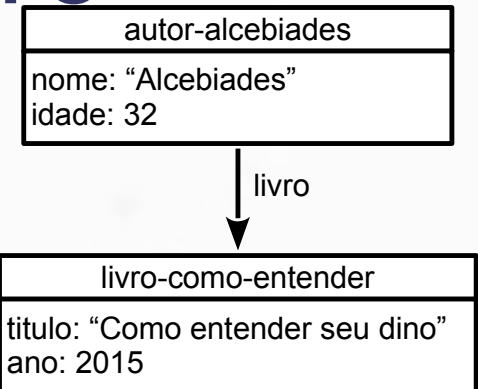


Referência Autor/Livro

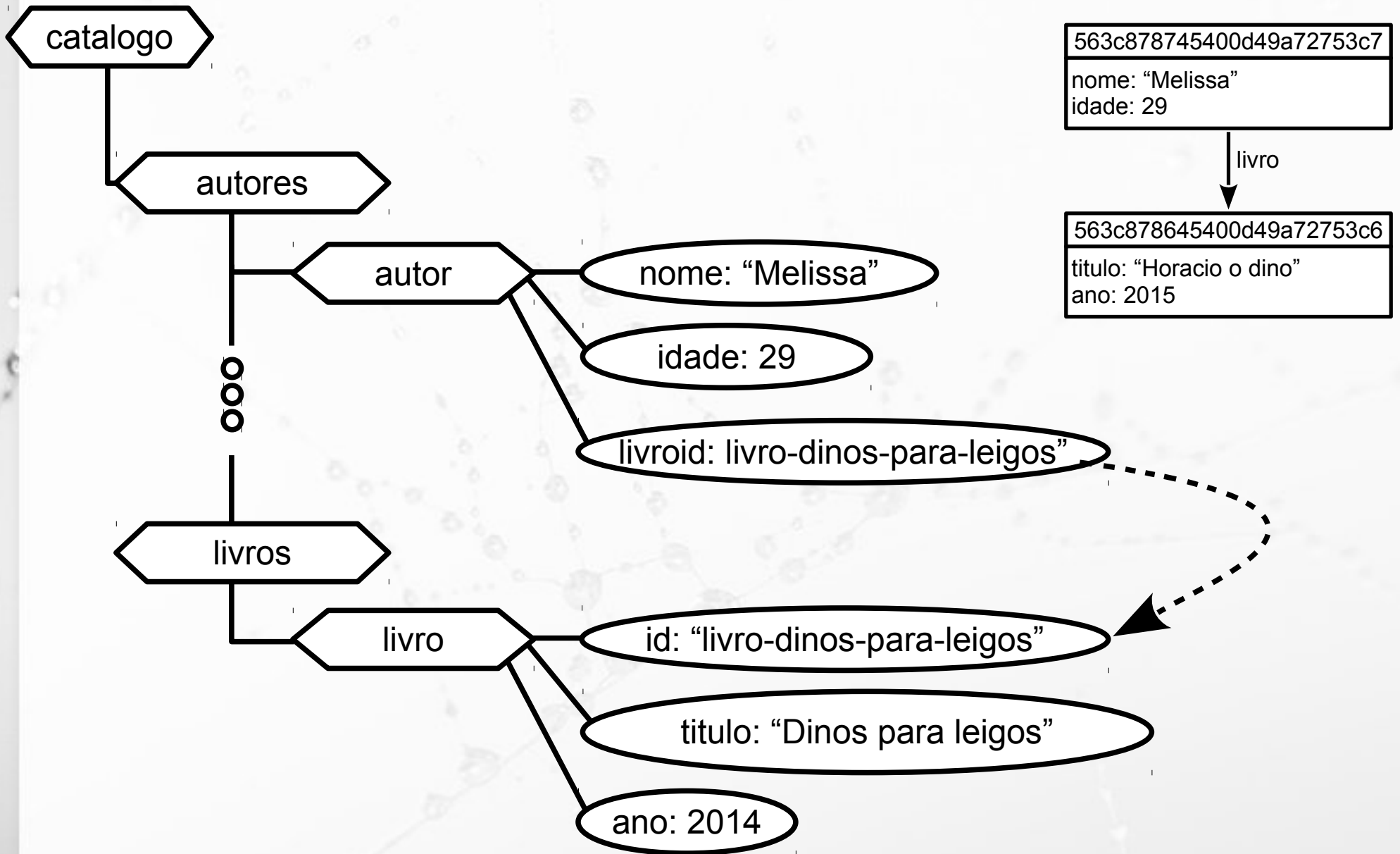


Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Alcebiades"
        idade="32"
        livroid="livro-como-entender" />
  ...
</autores>
...
<livros>
  <livro id="livro-como-entender"
        titulo="Como entender seu dino"
        ano="2015" />
  ...
</livros>
</catalogo>
```

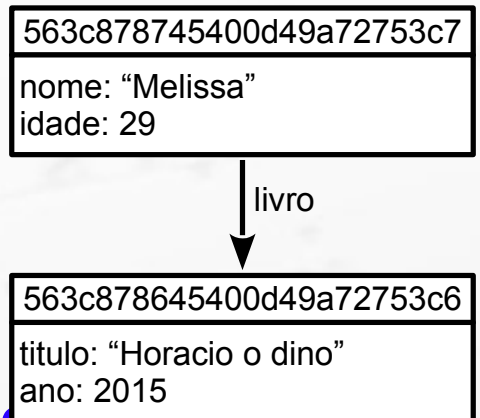


Referência Autor/Livro



Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Melissa"
        idade="29"
        livroid="livro-dinos-para-
leigos" />
  ...
</autores>
  ...
<livros>
  <livro id="livro-dinos-para-leigos"
        titulo="Dinos para leigo"
        ano="2014" />
  ...
</livros>
</catalogo>
```



Exercício 3 - Referência Cruzada

<http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-cruzada.xml>

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que nao?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```

XQuery com Embedded
<http://try.zorba.io/queries/xquery>

XQuery

let/return

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml')  
  
return $postsdoc
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que nao?</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
    <POST id="1512" data="2016-10-26">  
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    </POST>  
  </POST>  
</POST>  
</POST>  
</POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</POST>  
</POST>  
</POST>  
</FEED>
```

Exercício 4

Posts de 1º nível hierárquico

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  <POST id="1010" data="2016-10-25">
    <TEXT0>Por que nao?</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
</POST>
</POST>
</POST>
<POST id="1317" data="2016-10-10">
  <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  <POST id="1818" data="2016-10-18">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</POST>
</POST>
</FEED>
```

Exercício 4

Posts de 1º nível hierárquico

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml')  
return $postsdoc/FEED/POST
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que nao?</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
    <POST id="1512" data="2016-10-26">  
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    </POST>  
  </POST>  
</POST>  
</POST>  
</POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</POST>  
</POST>  
</POST>  
</FEED>
```

Exercício 5

Posts de qualquer nível hierárquico

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  <POST id="1010" data="2016-10-25">
    <TEXT0>Por que nao?</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
    <POST id="1512" data="2016-10-26">
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>
    </POST>
  </POST>
</POST>
</POST>
</POST>
<POST id="1317" data="2016-10-10">
  <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  <POST id="1818" data="2016-10-18">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</POST>
</POST>
</POST>
</FEED>
```

Exercício 5

Posts de qualquer nível hierárquico

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml')  
  
return $postsdoc//POST
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que nao?</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
    <POST id="1512" data="2016-10-26">  
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    </POST>  
  </POST>  
</POST>  
</POST>  
</POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</POST>  
</POST>  
</POST>  
</FEED>
```

Exercício 6

Número de posts

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  <POST id="1010" data="2016-10-25">
    <TEXT0>Por que nao?</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
</POST>
</POST>
</POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  <POST id="1818" data="2016-10-18">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</POST>
</POST>
</FEED>
```


Exercício 6

Número de posts

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml')  
  
return count($postsdoc//POST)
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que nao?</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
    <POST id="1512" data="2016-10-26">  
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    </POST>  
  </POST>  
</POST>  
</POST>  
<POST id="1317" data="2016-10-10">  
  <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</POST>  
</POST>  
</FEED>
```

Exercício 7

- A partir do esquema embedded XML, escreva uma sentença que indique quantos **reposts** diretos ou indiretos tem cada **post original**. Um **repost indireto** é quando há uma cadeia de referências que vai do repost ao post original.

Exercício 7

Reposts diretos ou indiretos

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded.xml')  
  
for $p in ($postsdoc/FEED/POST)  
return count($p//POST)
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que nao?</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
    <POST id="1512" data="2016-10-26">  
      <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    </POST>  
  </POST>  
</POST>  
</POST>  
</POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</POST>  
</POST>  
</POST>  
</FEED>
```

Exercício ao Contrário

Embedded 1:1

cada post com 1 repost

<http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-1-1.xml>

```
<FEED>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>
  <POST id="1010" data="2016-10-25">
    <TEXT0>Por que não?</TEXT0>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>
  </POST>
</POST>
</POST>
<POST id="1818" data="2016-10-18">
  <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
<POST id="2021" data="2016-10-15">
  <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  <POST id="1317" data="2016-10-10">
    <TEXT0>0 professor disse que o Plessiosauro não é um
dinossauro.</TEXT0>
  </POST>
</POST>
</POST>
</FEED>
```

not()

Posts sem Nós Filhos

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded-1-1.xml')  
return $postsdoc//POST[not(POST)]
```

```
<FEED>  
  <POST id="1512" data="2016-10-26">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
  <POST id="2625" data="2016-10-25">  
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>  
  <POST id="1010" data="2016-10-25">  
    <TEXT0>Por que não?</TEXT0>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>  
  </POST>  
</POST>  
</POST>  
</POST>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  <POST id="2021" data="2016-10-15">  
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>0 professor disse que o Plessiosauro não é um dinossauro.</TEXT0>  
  </POST>  
</POST>  
</POST>  
</FEED>
```

::ancestor

Todos os ancestrais de um nó

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded-1-1.xml')  
return $postsdoc//POST[not(POST)]/ancestor::*
```

```
<FEED>  
  <POST id="1512" data="2016-10-26">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    <POST id="2625" data="2016-10-25">  
      <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>  
      <POST id="1010" data="2016-10-25">  
        <TEXT0>Por que não?</TEXT0>  
        <POST id="1513" data="2016-10-24">  
          <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>  
        </POST>  
      </POST>  
    </POST>  
  </POST>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
    <POST id="2021" data="2016-10-15">  
      <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
      <POST id="1317" data="2016-10-10">  
        <TEXT0>O professor disse que o Plessiossauro não é um dinossauro.</TEXT0>  
      </POST>  
    </POST>  
  </POST>  
</FEED>
```

::ancestor

Quantos ancestrais tem cada nó?

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded-1-1.xml')  
  
for $p in ($postsdoc//POST[not(POST)])  
return count($p/ancestor::*)-1
```

```
<FEED>  
  <POST id="1512" data="2016-10-26">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    <POST id="2625" data="2016-10-25">  
      <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>  
      <POST id="1010" data="2016-10-25">  
        <TEXT0>Por que não?</TEXT0>  
        <POST id="1513" data="2016-10-24">  
          <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>  
        </POST>  
      </POST>  
    </POST>  
  </POST>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
    <POST id="2021" data="2016-10-15">  
      <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>  
      <POST id="1317" data="2016-10-10">  
        <TEXT0>0 professor disse que o Plessiossauro não é um dinossauro.</TEXT0>  
      </POST>  
    </POST>  
  </POST>  
</FEED>
```


Trabalhando com Referência Cruzada

Referência Cruzada XQuery

```
let $postsdoc:=doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-  
cruzada.xml')  
  
return $postsdoc
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  </POST>  
  <POST id="1010" data="2016-10-25" ref_post="1513">  
    <TEXT0>Por que nao?</TEXT0>  
  </POST>  
  <POST id="2625" data="2016-10-25" ref_post="1010">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
  </POST>  
  <POST id="1512" data="2016-10-26" ref_post="2625">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
  </POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>  
  </POST>  
  <POST id="2021" data="2016-10-15" ref_post="1317">  
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
  </POST>  
  <POST id="1818" data="2016-10-18" ref_post="2021">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
  </POST>  
</FEED>
```

Relembrando o Join

- Publicações associadas à categoria cujo <label> em inglês seja 'e-Science Domain'. A associação entre o label e a key da categoria deve ser feita na consulta.

```
for $c in ($publicadoc//categories/category),  
    $p in ($publicadoc//publication)  
where $c/label[@lang='en-US'] = 'e-Science  
Domain' and  
    $p/key = $c/@key  
return $p
```

Exercício 8

Listas id de posts e respectivos reposts

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que nao?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```

Exercício 8

Listas id de posts e respectivos reposts

```
let $postsdoc:=doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-cruzada.xml')
for $p1 in ($postsdoc//POST),
    $p2 in ($postsdoc//POST)
where $p1/@ref_post = $p2/@id
return {concat('Post: ', data($p1/@id), ', ', 'Repost: ', data($p2/@id))}
```

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que nao?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```

Gerando novo XML com If

Gerando novo XML com If Diferenciando <post> de <repost>

A partir do XML gere a seguinte saída:

```
<REPOST>1512</REPOST>  
<REPOST>2625</REPOST>  
<REPOST>1010</REPOST>  
<POST>1513</POST>  
<REPOST>1818</REPOST>  
<REPOST>2021</REPOST>  
<POST>1317</POST>
```

```
<FEED>  
  <POST id="1513" data="2016-10-24">  
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>  
  </POST>  
  <POST id="1010" data="2016-10-25" ref_post="1513">  
    <TEXT0>Por que nao?</TEXT0>  
  </POST>  
  <POST id="2625" data="2016-10-25" ref_post="1010">  
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>  
  </POST>  
  <POST id="1512" data="2016-10-26" ref_post="2625">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
  </POST>  
  <POST id="1317" data="2016-10-10">  
    <TEXT0>O professor disse que o Plessiosauro nao e um dinossauro.</TEXT0>  
  </POST>  
  <POST id="2021" data="2016-10-15" ref_post="1317">  
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
  </POST>  
  <POST id="1818" data="2016-10-18" ref_post="2021">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
```

Gerando novo XML com If <post> de <repost> atributos

```
let $postsdoc:=doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-cruzada.xml')

for $p in ($postsdoc//POST)
return if ($p[@ref_post])
  then <REPOST>{data($p/@id)}</REPOST>
  else <POST>{data($p/@id)}</POST>
```

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que nao?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>0 professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>0 que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```


De volta ao Embedded 1:1

Gerando novo XML com If Diferenciando <post> de <repost>

A partir do XML gere a seguinte saída:

```
<REPOST>1512</REPOST>
<REPOST>2625</REPOST>
<REPOST>1010</REPOST>
<POST>1513</POST>
<REPOST>1818</REPOST>
<REPOST>2021</REPOST>
<POST>1317</POST>
```

```
<FEED>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>
    <POST id="1010" data="2016-10-25">
      <TEXT0>Por que não?</TEXT0>
      <POST id="1513" data="2016-10-24">
        <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>
      </POST>
    </POST>
  </POST>
</POST>
<POST id="1818" data="2016-10-18">
  <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
    <POST id="1317" data="2016-10-10">
      <TEXT0>O professor disse que o Plessiosauro não é um dinossauro.</TEXT0>
    </POST>
  </POST>
</POST>
```

Gerando novo XML com If Diferenciando <post> de <repost>

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded-1-1.xml')

for $p in ($postsdoc//POST)
return if ($p[not(POST)])
  then <POST>{data($p/@id)}</POST>
  else <REPOST>{data($p/@id)}</REPOST>
```

```
<FEED>
  <POST id="1512" data="2016-10-26">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>
    <POST id="1010" data="2016-10-25">
      <TEXT0>Por que não?</TEXT0>
    <POST id="1513" data="2016-10-24">
      <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>
    </POST>
  </POST>
</POST>
</POST>
<POST id="1818" data="2016-10-18">
  <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
    <POST id="1317" data="2016-10-10">
      <TEXT0>O professor disse que o Plessiosauro não é um dinossauro.</TEXT0>
    </POST>
  </POST>
</POST>
</POST>
</FEED>
```

Gerando novo XML com If <post> de <repost> atributos

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social-embedded-1-1.xml')  
  
for $p in ($postsdoc//POST)  
return if ($p[not(POST)])  
  then <POST id="{data($p/@id)}"></POST>  
  else <REPOST id="{data($p/@id)}"></REPOST>
```

```
<FEED>  
  <POST id="1512" data="2016-10-26">  
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>  
    <POST id="2625" data="2016-10-25">  
      <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>  
      <POST id="1010" data="2016-10-25">  
        <TEXT0>Por que não?</TEXT0>  
        <POST id="1513" data="2016-10-24">  
          <TEXT0>Dinossauros não podem ter se tornado pássaros.</TEXT0>  
        </POST>  
      </POST>  
    </POST>  
  </POST>  
  <POST id="1818" data="2016-10-18">  
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>  
    <POST id="2021" data="2016-10-15">  
      <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>  
      <POST id="1317" data="2016-10-10">  
        <TEXT0>O professor disse que o Plessiosauro não é um dinossauro.</TEXT0>  
      </POST>  
    </POST>  
  </POST>  
</FEED>
```

Transformando Embedded 1:1 em Referência Cruzada

Exercício 9

Transforme Embedded em Ref. Cruzada

```
<FEED>
  <POST id="1512" data="2016-10-24">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1512">
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="2625">
    <TEXT0>Por que não?</TEXT0>
  </POST>
  <POST id="1513" data="2016-10-25" ref_post="1010">
    <TEXT0>Dinossauros não podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-26" ref_post="1513">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro não é um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro não é um dinossauro.</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros não podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que não?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Não dá pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro não é um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
  </POST>
</FEED>
```

Exercício 9

Transforme Embedded em Ref. Cruzada

```
let $postsdoc := doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/post-rede-social.xml')

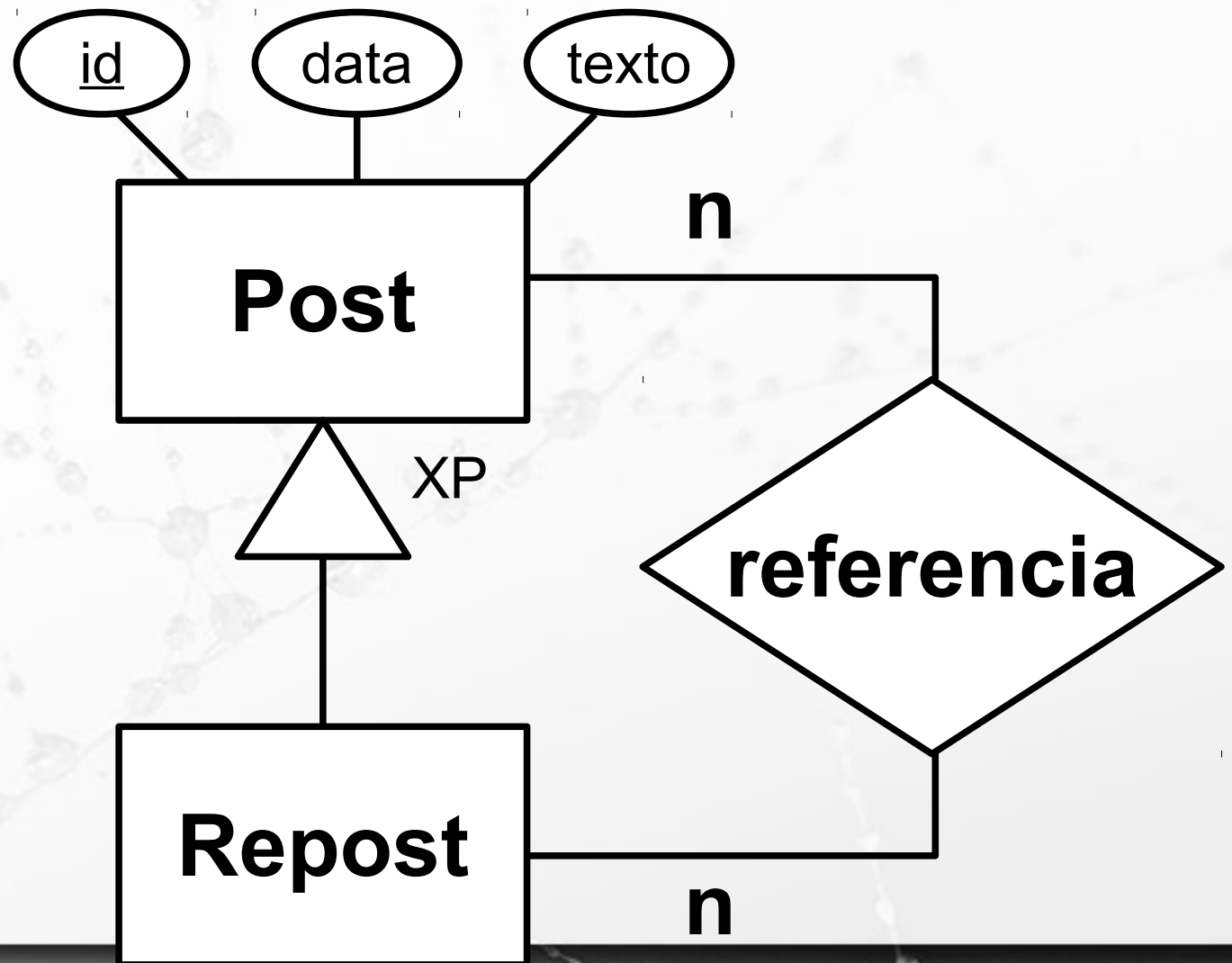
for $p in ($postsdoc//POST)
return if ($p[not(POST)])
  then <POST id="{data($p/@id)}"
data="{data($p/@data)}"><TEXT0>{data($p/TEXT0/text())}</TEXT0></POST>
  else <POST id="{data($p/@id)}" data="{data($p/@data)}"
ref_post="{data($p/POST/@id)}"><TEXT0>{data($p/TEXT0/text())}</TEXT0></POST>
```

```
<FEED>
  <POST id="1512" data="2016-10-24">
    <TEXT0>De fato, ele ia perde</TEXT0>
  <POST id="2625" data="2016-10-25">
    <TEXT0>Não dá pra imaginar</TEXT0>
    <POST id="1010" data="2016-10-25">
      <TEXT0>Por que não?</TEXT0>
    <POST id="1513" data="2016-10-24">
      <TEXT0>Dinossauros n</TEXT0>
    </POST>
  </POST>
</POST>
<POST id="1818" data="2016-10-18">
  <TEXT0>Talvez melhore seu en</TEXT0>
  <POST id="2021" data="2016-10-15">
    <TEXT0>O que isso tem a ve</TEXT0>
    <POST id="1317" data="2016-10-10">
      <TEXT0>O professor dis</TEXT0>
    </POST>
  </POST>
</POST>
```

```
<FEED>
  <POST id="1513" data="2016-10-24">
    <TEXT0>Dinossauros nao podem ter se tornado passaros.</TEXT0>
  </POST>
  <POST id="1010" data="2016-10-25" ref_post="1513">
    <TEXT0>Por que nao?</TEXT0>
  </POST>
  <POST id="2625" data="2016-10-25" ref_post="1010">
    <TEXT0>Nao da pra imaginar um Tiranossauro com penas.</TEXT0>
  </POST>
  <POST id="1512" data="2016-10-26" ref_post="2625">
    <TEXT0>De fato, ele ia perder a moral!</TEXT0>
  </POST>
  <POST id="1317" data="2016-10-10">
    <TEXT0>O professor disse que o Plessiossauro nao e um dinossauro.</TEXT0>
  </POST>
  <POST id="2021" data="2016-10-15" ref_post="1317">
    <TEXT0>O que isso tem a ver com Banco de Dados?</TEXT0>
  </POST>
  <POST id="1818" data="2016-10-18" ref_post="2021">
    <TEXT0>Talvez melhore seu entendimento de XML Schema?</TEXT0>
```

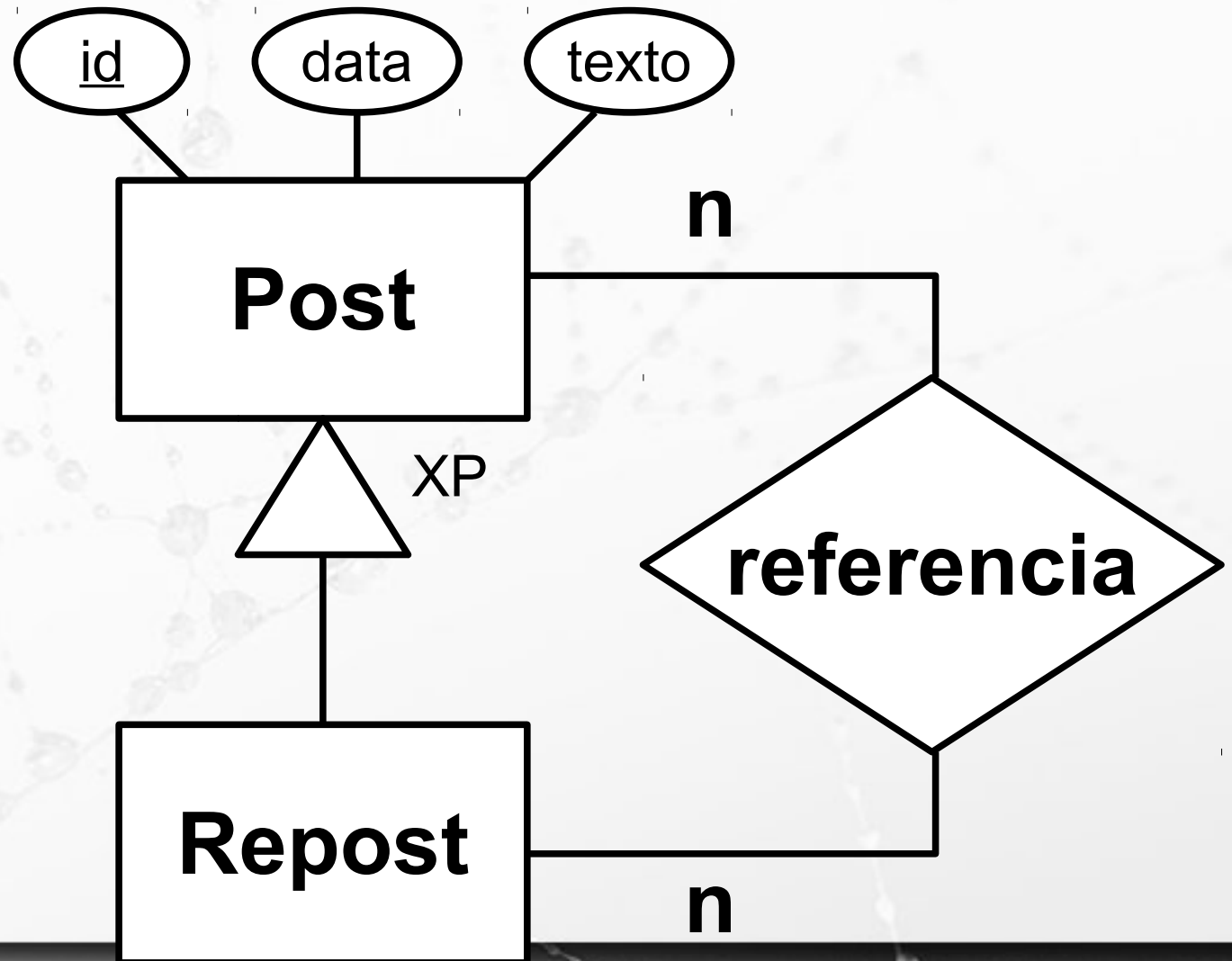
Exercício 10

- Elabore um esquema relacional que atenda a este modelo conceitual.



Exercício 11

- Escreva um exemplo de um arquivo XML que seja capaz de representar o modelo conceitual.





Elementos e Atributos

- **Atributos:**

```
<autor cpf="487.526.548-74" nascimento="12/5/1960"> Horácio </autor>
```

- **Elementos vazios:**

```
<esgotado/>
```

- *Links* para elementos (#):

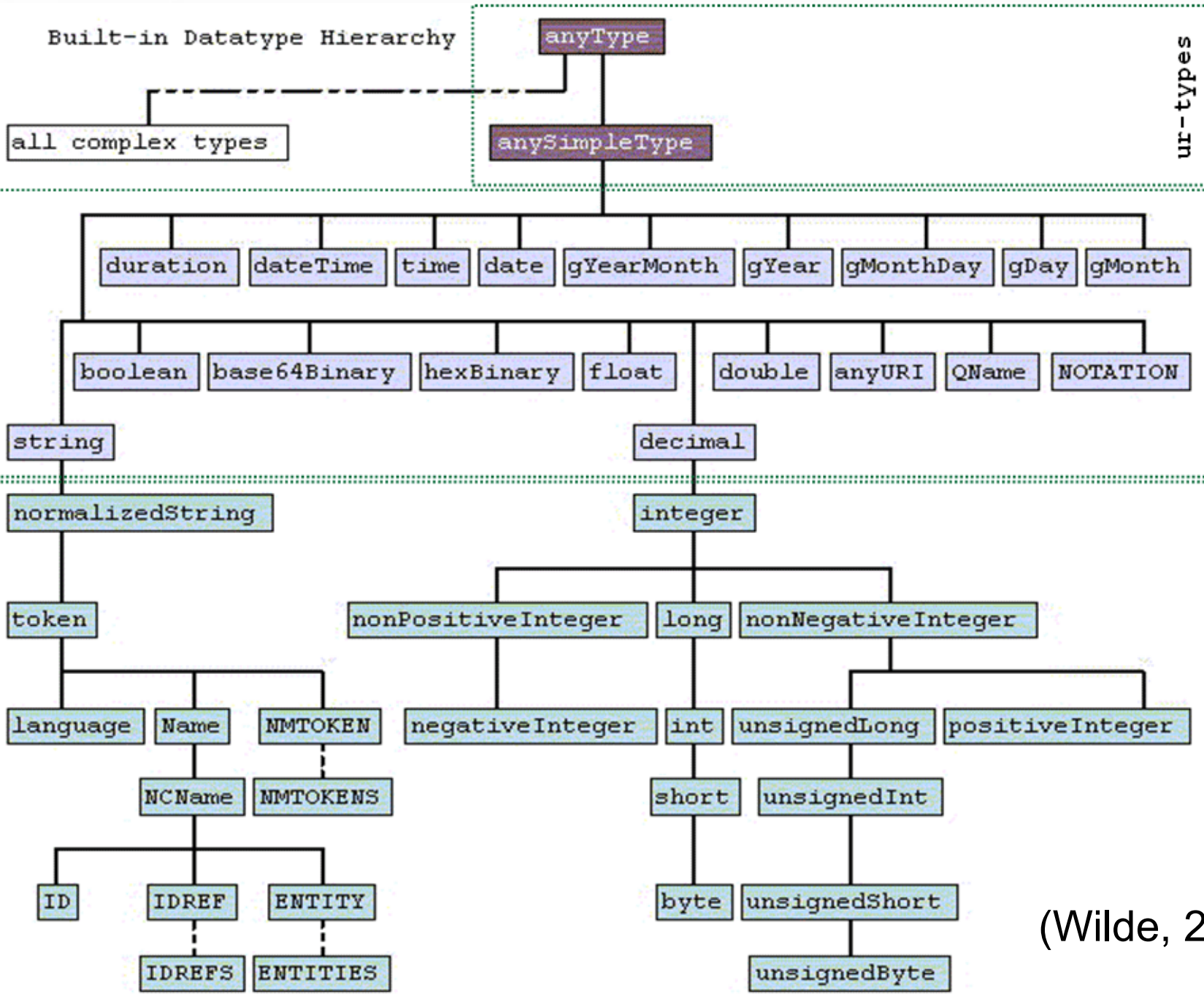
```
http://www.dominio.org/documento.html#bibliografia
```

- HTML usa esta estratégia em links para fragmentos.

XML Schema

Tipos de Dados

Built-in Datatype Hierarchy



ur-types

primitive types

derived types

- ur types
- built-in primitive types
- built-in derived types
- derived by restriction
- derived by list
- derived by extension or

(Wilde, 2006)

Estruturado x Semi-estruturado

- Estruturado

- formato estrito
- e.g., modelo relacional
- cada registro segue o mesmo formato

(Elmasri, 2010)

- Semi-estruturado

- itens de dados podem ter estruturas variadas
- grupos de itens compartilham estruturas

Query

- **XPath**

- Especifica expressões na forma de caminhos que atendem padrões para alcançar nós específicos (elementos ou atributos)

- **XQuery**

- Queries para XML (usam XPath)

XPath

/	no começo → nó raiz entre nós → separador hierárquico
//	no começo → nó que começa em qualquer nível hierárquico entre nós → separador hierárquico
@atr	indica um atributo
*	qualquer elemento
@*	qualquer atributo
text()	conteúdo textual de um nó

Xpath Exemplos

<http://www.online-toolz.com/tools/xpath-editor.php>

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

/fichario/individuo

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

/fichario/individuo

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
<individuo nome="Asdrubal da Silva">
<idade>15</idade>
<genero>masculino</genero>
</individuo>
```

```
-----
<individuo nome="Quincas Borba">
<idade>33</idade>
<genero>masculino</genero>
</individuo>
```

```
-----
<individuo nome="Doriana Margarina">
<idade>42</idade>
<genero>feminino</genero>
</individuo>
```

Xpath

Exemplos

//individuo

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

//individuo

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
<individuo nome="Asdrubal da Silva">
<idade>15</idade>
<genero>masculino</genero>
</individuo>
-----
<individuo nome="Quincas Borba">
<idade>33</idade>
<genero>masculino</genero>
</individuo>
-----
<individuo nome="Doriana Margarina">
<idade>42</idade>
<genero>feminino</genero>
</individuo>
```

Xpath

Exemplos

```
//individuo/@nome
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

```
//individuo/@nome
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

nome="Asdrubal da Silva"

nome="Quincas Borba"

nome="Doriana Margarina"

Xpath

Exemplos

/fichario/*/idade

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```


Xpath

Exemplos

/fichario/*/idade

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
<idade>15</idade>
```

```
-----
<idade>33</idade>
```

```
-----
<idade>42</idade>
```

Xpath

Exemplos

/fichario/*/idade

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

/fichario/*/idade/text()

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

15

33

42

XPath

Predicados

[]	predicado – filtra o conjunto de nós
[elem]	filtra nós com elemento
[@atr]	filtra nós com atributo
[n]	filtra enésimo nó
[last()]	filtra último nó
[@atr=val] [@atr>val] [elem=val] [elem>val]	filtra nós que atendem a condição

Xpath

Exemplos

```
//individuo[2]
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

```
//individuo[2]
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<individuo nome="Quincas Borba">  
<idade>33</idade>  
<genero>masculino</genero>  
</individuo>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
<individuo nome="Quincas Borba">
  <idade>33</idade>
  <genero>masculino</genero>
</individuo>
```


Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<idade>33</idade>
```

Xpath Exemplos

```
//individuo[@nome="Quincas Borba"]/idade/text()
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

Xpath

Exemplos

```
//individuo[@nome="Quincas Borba"]/idade/text()
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

33

Xpath

Exemplos

```
//individuo[idade>20]/@nome
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

Xpath Exemplos

```
//individuo[idade>20]/@nome
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
nome="Quincas Borba"  
-----  
nome="Doriana Margarina"
```

XQuery

FOR <variable bindings to individual nodes (elements)>

LET <variable bindings to collections of nodes (elements)>

WHERE <qualifier conditions>

RETURN <query result specification>

(Elmasri, 2011)

XQuery Exemplos

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{$message}</livro>
```


XQuery Exemplos

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{$message}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Dinotopia</livro>
```

XML no resultado

- `{}` → para indicar valores de retorno dentro do XML
- `data()` → extrai o conteúdo literal do elemento ou atributo

XQuery

Label parte do XML externo

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return data($message)
```

```
<?xml version="1.0" encoding="UTF-8"?>  
Dinotopia
```

XQuery

Label parte do XML externo

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>Título: {$message}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

XQuery

Label dentro da { }

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{data('Título:'),  
$message}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

- vírgula concatena com espaços intermediários.

XQuery concat()

<http://try.zorba.io/>

```
xquery version "1.0";  
let $message := 'Dinotopia'  
return  
<livro>{concat('Título: ',  
$message)}</livro>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<livro>Título: Dinotopia</livro>
```

- `concat()` concatena sem espaços intermediários.

XQuery XML Base

<http://www.ic.unicamp.br/~santanch/teaching/db/xml/fichario.xml>

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

XQuery XML Base

`http://www.ic.unicamp.br/~santanch/teaching/db/xml/fichario.xml`

- Para simplificar vou chamá-lo de

`icunicamp:fichario.xml`

XQuery

let/return

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario/individuo/idade
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc/fichario/individuo/idade
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<idade>15</idade>  
<idade>33</idade>  
<idade>42</idade>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc//individuo[idade>20][genero="masculino"]
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery

Xquery e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return $fichariodoc//individuo[idade>20][genero="masculino"]
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<individuo nome="Quincas Borba">  
  <idade>33</idade>  
  <genero>masculino</genero>  
</individuo>
```

XQuery count()

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return count($fichariodoc//individuo)
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery count()

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
return count($fichariodoc//individuo)
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

3

XQuery

count() e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
return count($fichariodoc//individuo[idade>20]  
[genero="masculino"])
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```


XQuery count() e XPath

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
return count($fichariodoc//individuo[idade>20]  
[genero="masculino"])
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

1

XQuery for

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return $i/idade
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery for

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return $i/idade
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<idade>15</idade>  
<idade>33</idade>  
<idade>42</idade>
```

XQuery where

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return $i
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery where

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return $i
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<individuo nome="Quincas Borba">  
  <idade>33</idade>  
  <genero>masculino</genero>  
</individuo>  
<individuo nome="Doriana Margarina">  
  <idade>42</idade>  
  <genero>feminino</genero>  
</individuo>
```

XQuery

XML no resultado

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery

XML no resultado

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery

XML no resultado

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```


XQuery

XML no resultado

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```

XQuery

Xquery dentro de outro XQuery

```
let $fichariodoc := doc('icunicamp:fichario.xml')
return
<classificacao>
{
  for $i in ($fichariodoc//individuo)
  where $i[idade>17]
  return <maior>{data($i/@nome)}</maior>
}
</classificacao>
```

```
<fichario>
  <individuo nome="Asdrubal da Silva">
    <idade>15</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Quincas Borba">
    <idade>33</idade>
    <genero>masculino</genero>
  </individuo>
  <individuo nome="Doriana Margarina">
    <idade>42</idade>
    <genero>feminino</genero>
  </individuo>
</fichario>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<classificacao>
  <maior>Quincas Borba</maior>
  <maior>Doriana Margarina</maior>
</classificacao>
```

XQuery order by

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
order by $i/@nome  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

XQuery order by

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
where $i[idade>17]  
order by $i/@nome  
return <maior>{data($i/@nome)}</maior>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<maior>Doriana Margarina</maior>  
<maior>Quincas Borba</maior>
```

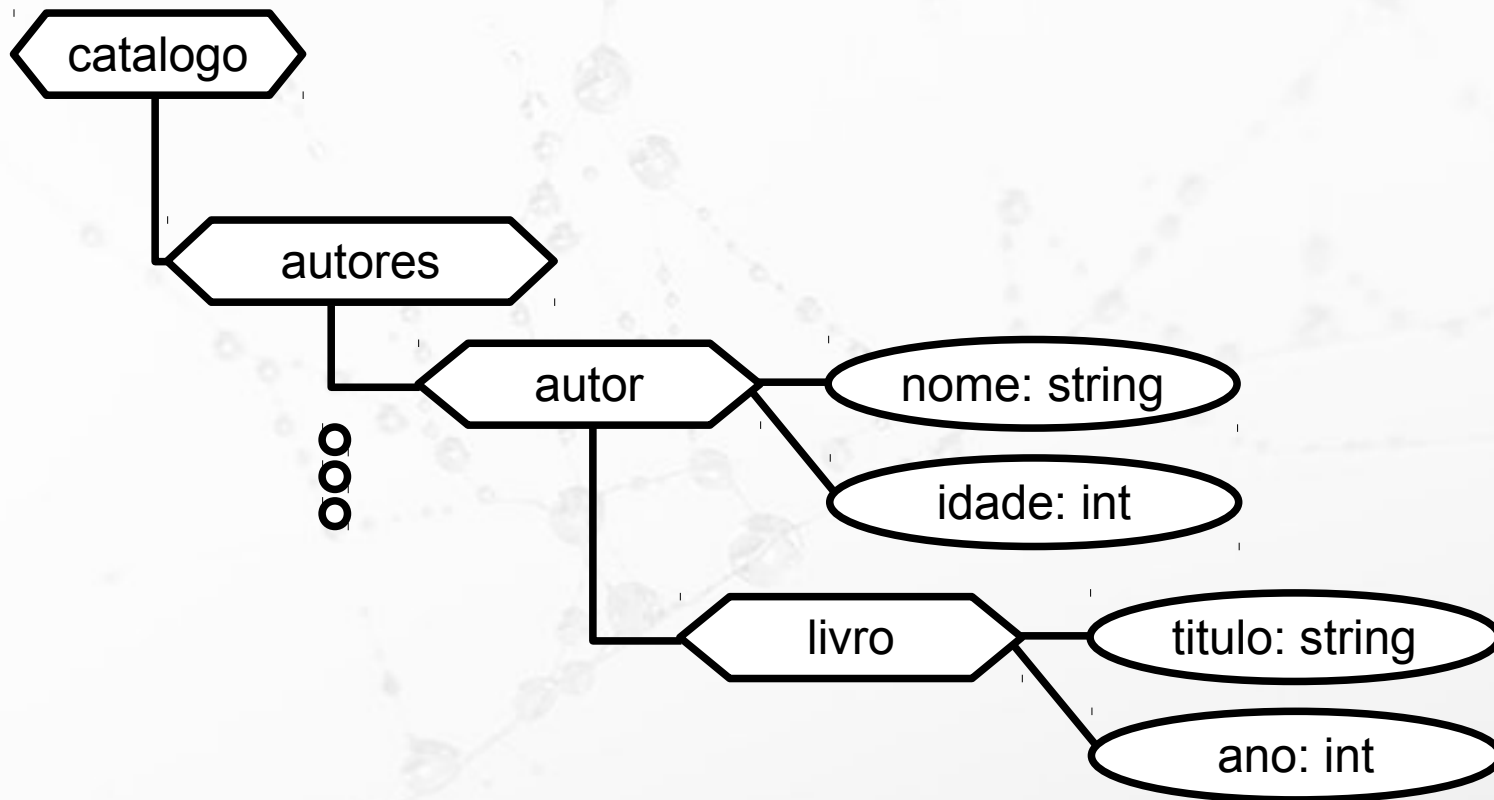
XQuery if

```
let $fichariodoc := doc('icunicamp:fichario.xml')  
  
for $i in ($fichariodoc//individuo)  
return if ($i[idade>=18])  
  then <maior>{data($i/@nome)}</maior>  
  else <menor>{data($i/@nome)}</menor>
```

```
<fichario>  
  <individuo nome="Asdrubal da Silva">  
    <idade>15</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Quincas Borba">  
    <idade>33</idade>  
    <genero>masculino</genero>  
  </individuo>  
  <individuo nome="Doriana Margarina">  
    <idade>42</idade>  
    <genero>feminino</genero>  
  </individuo>  
</fichario>
```

```
<menor>Asdrubal da Silva</menor>  
<maior>Quincas Borba</maior>  
<maior>Doriana Margarina</maior>
```

Autor/Livro Embedded



Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Doriana">
```

```
    <livro titulo="Horacio o dino"  
ano="2015" />
```

```
  </autor>
```

```
  ...
```

```
</autores>
```

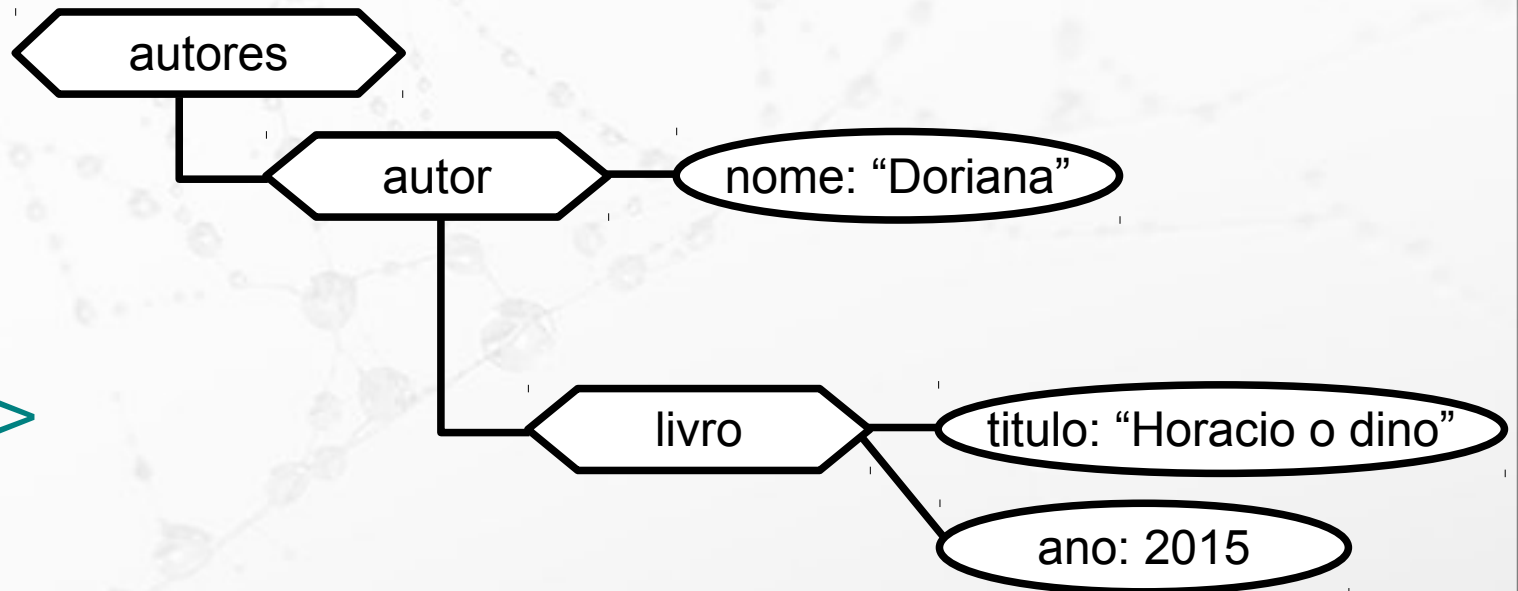
```
...
```

```
</catalogo>
```

563c871c45400d49a72753c4
nome: "Doriana"

livro

titulo: "Horacio o dino" ano: 2015



Livro Embedded em Autor

```
<catalogo>
```

```
<autores>
```

```
  <autor nome="Asdrubal" idade="25"
```

```
    <livro titulo="Vida sentimental dos dinossauros"  
ano="2013" />
```

```
  </autor>
```

```
...
```

```
</autores>
```

```
...
```

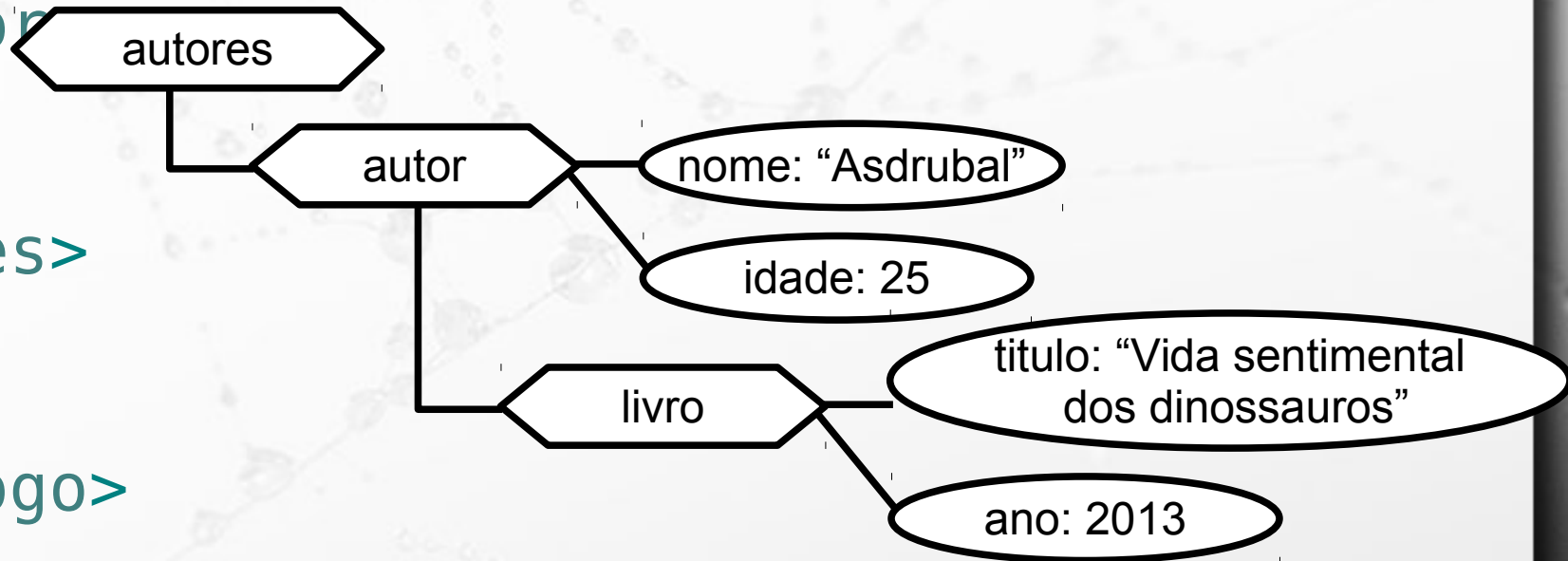
```
</catalogo>
```

563c873745400d49a72753c5

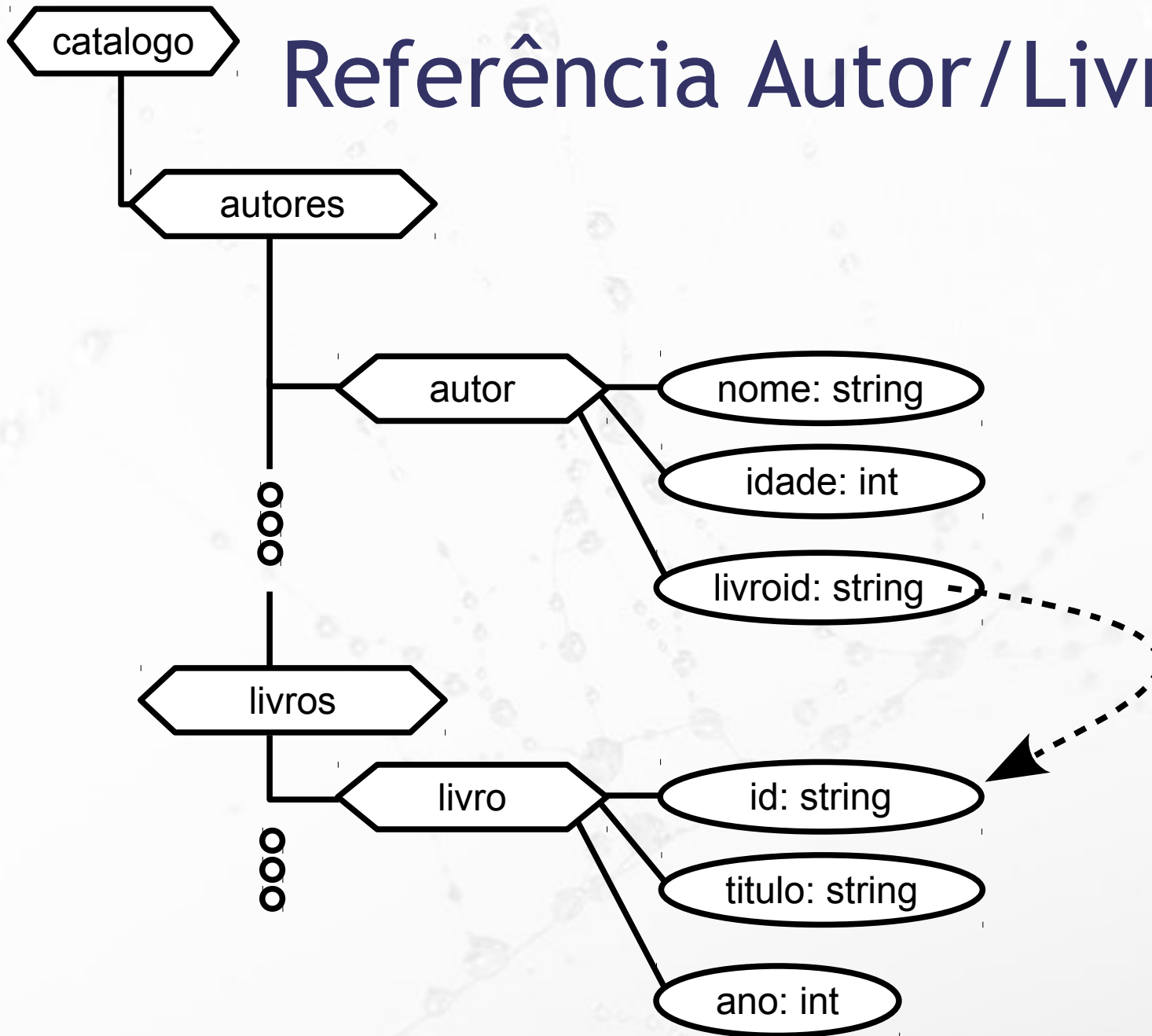
nome: "Asdrubal"
idade: 25

livro

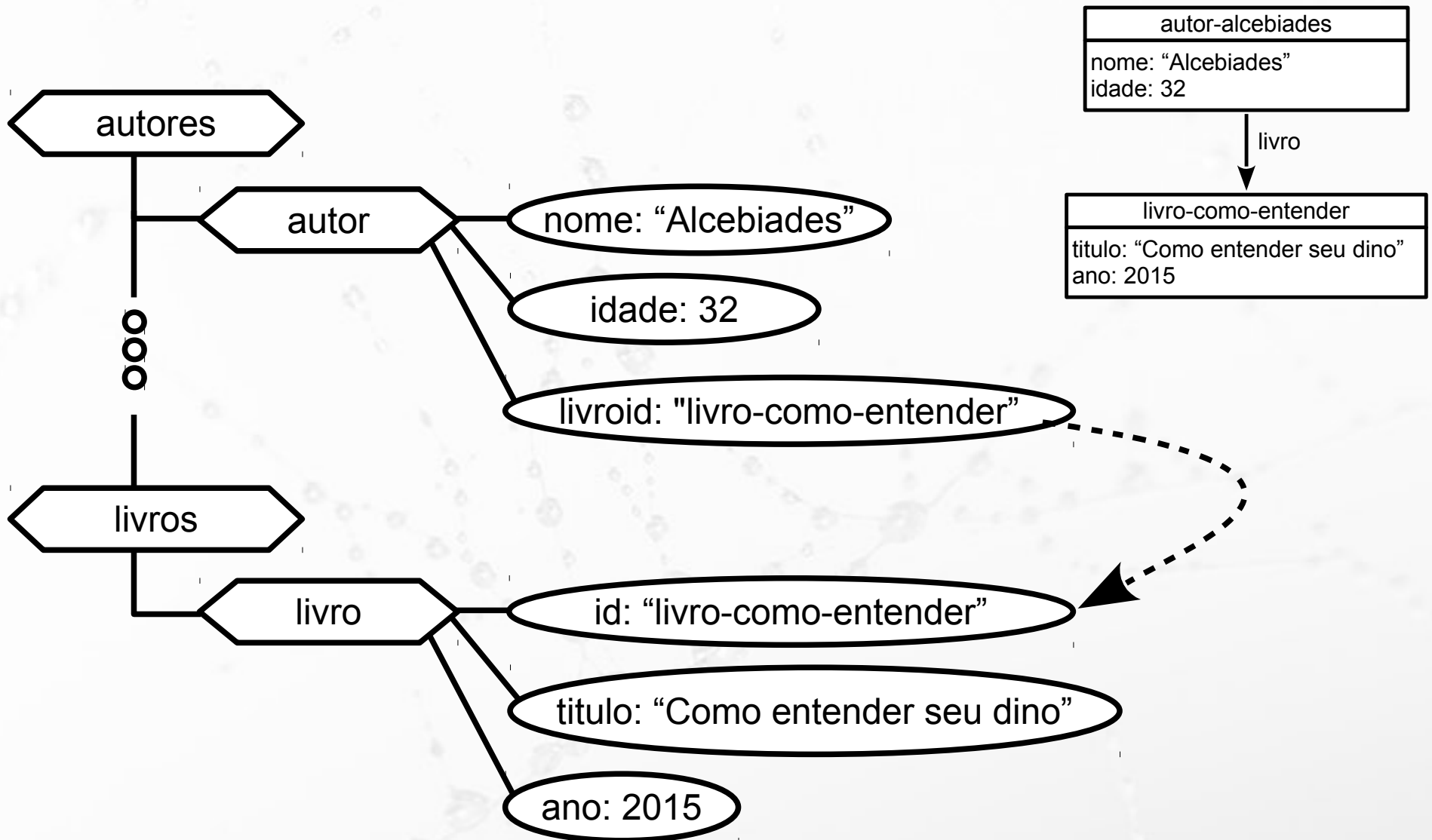
titulo: "Vida sentimental dos dinossauros"
ano: 2013



Referência Autor/Livro

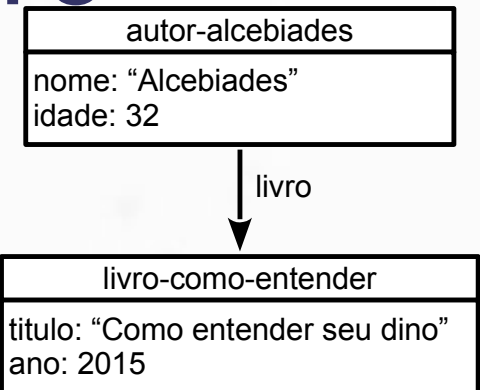


Referência Autor/Livro

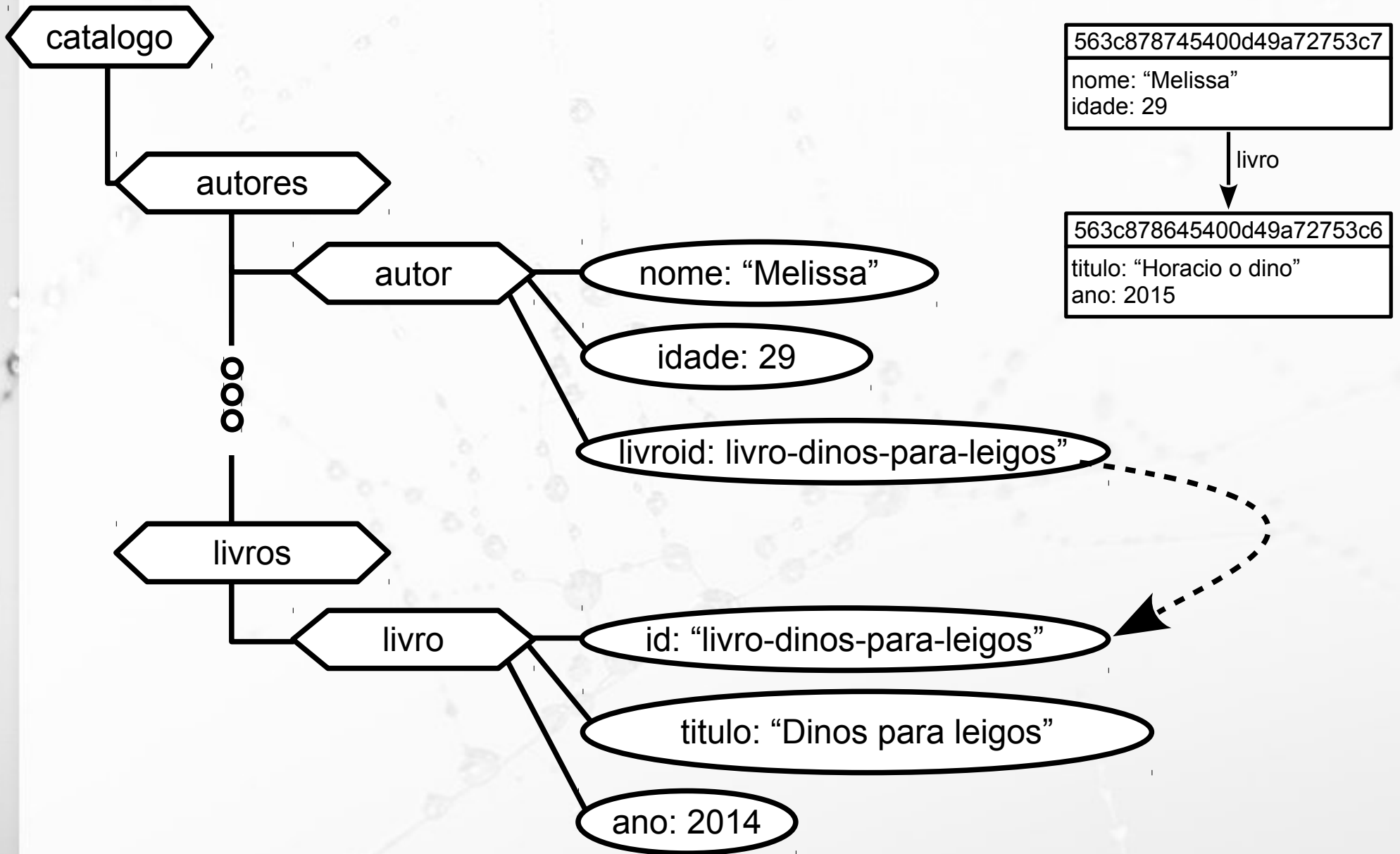


Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Alcebiades"
        idade="32"
        livroid="livro-como-entender" />
  ...
</autores>
  ...
<livros>
  <livro id="livro-como-entender"
        titulo="Como entender seu dino"
        ano="2015" />
  ...
</livros>
</catalogo>
```

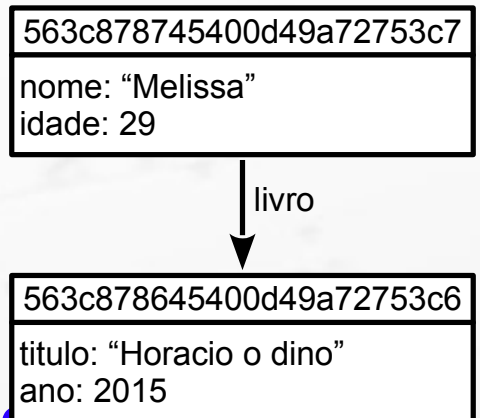


Referência Autor/Livro



Referência Autor/Livro

```
<catalogo>
<autores>
  <autor nome="Melissa"
        idade="29"
        livroid="livro-dinos-para-
leigos" />
  ...
</autores>
...
<livros>
  <livro id="livro-dinos-para-leigos"
        titulo="Dinos para leigo"
        ano="2014" />
  ...
</livros>
</catalogo>
```



XQuery Join

- Retorne o nome dos autores e o título de seus livros:

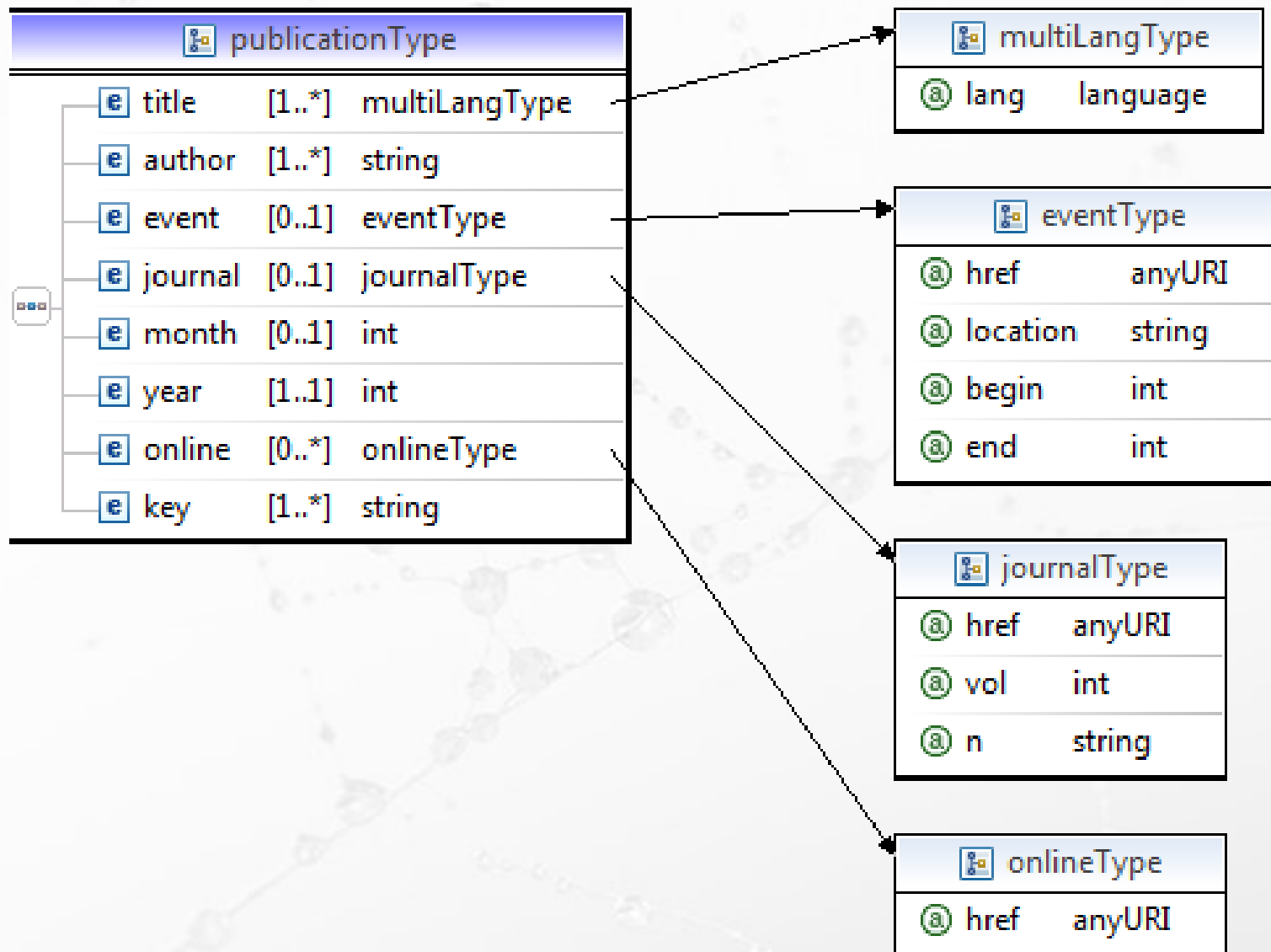
```
let $autorlivro :=  
doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/autor-  
livro-referencia.xml')  
  
for $a in ($autorlivro/catalogo/autores/autor),  
    $l in ($autorlivro/catalogo/livros/livro)  
where $a/@livroid = $l/@id  
  
return {concat('Autor: ', data($a/@nome), ', Livro: ',  
data($l/@titulo))}
```

XQuery group by

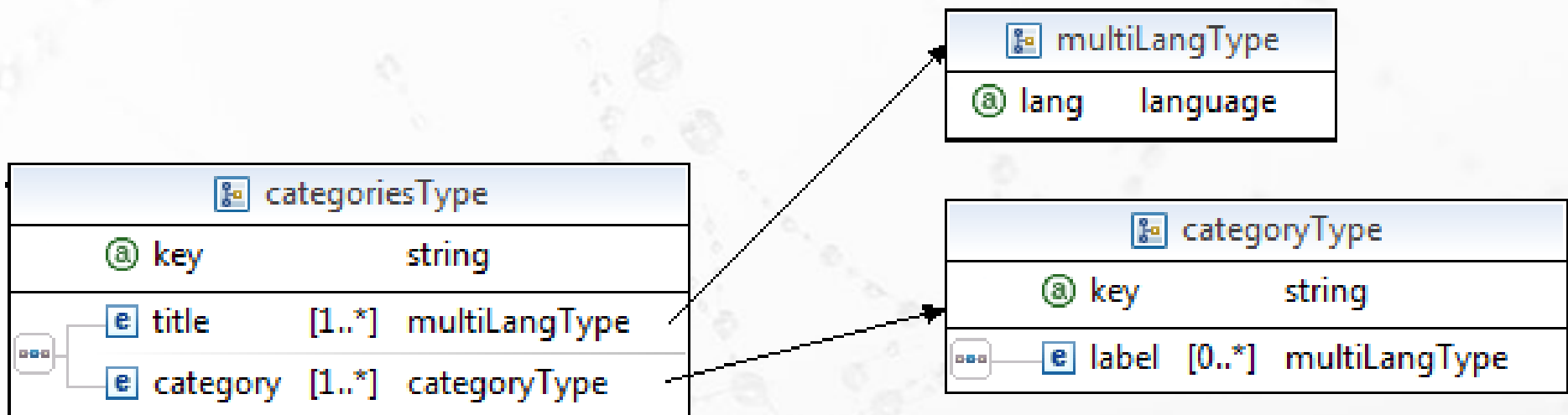
- Retorne o número de livros por ano:

```
let $autorlivro :=  
doc('http://www.ic.unicamp.br/~santanch/teaching/db/xml/autor-livro-referencia.xml')  
  
for $l in ($autorlivro/catalogo/livros/livro)  
group by $l/@ano  
return {'Livros -- ano: ', 'quantidade: ', count($l)}
```

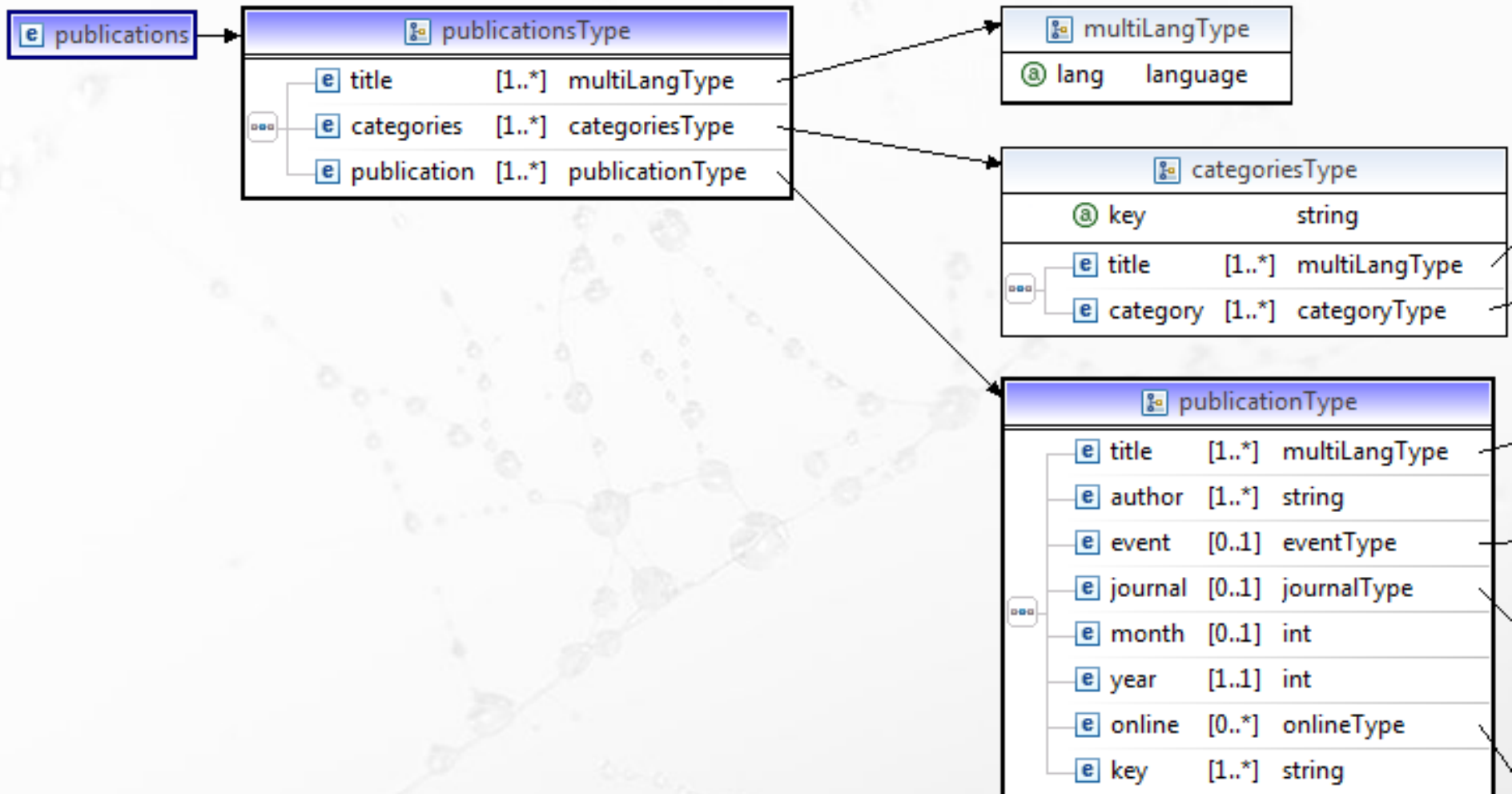
Publication Type



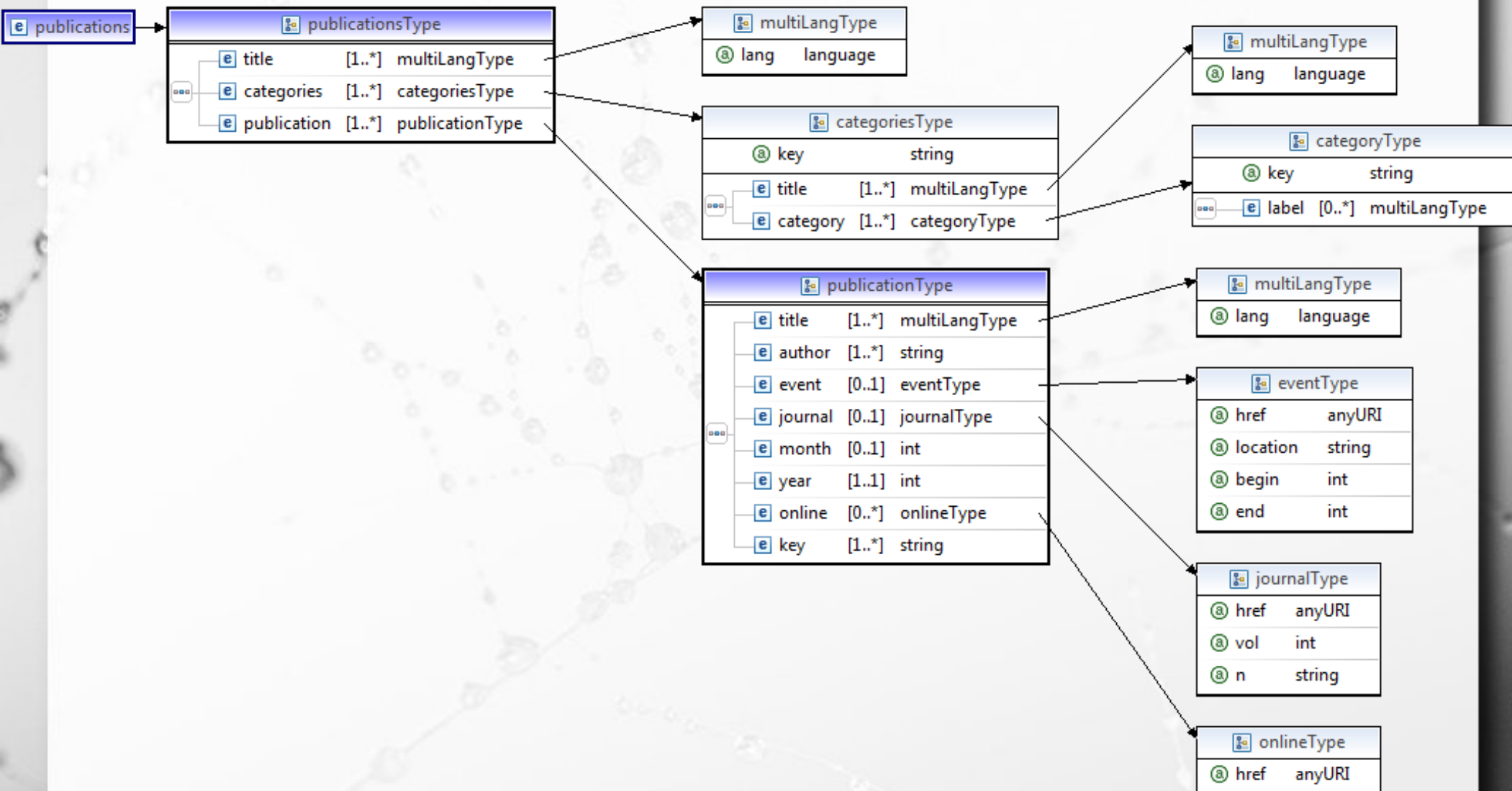
CategoryType and CategoriesType



publicationsType



Everything together



Questão

- Retorne quantas publicações são posteriores ao ano de 2011

Questão

- Retorne quantas publicações são posteriores ao ano de 2011
- `return
count($publicadoc // publication[year>2010])`

Questão

- Retorne a categoria cujo <label> em inglês seja 'e-Science Domain'

Questão

- Retorne a categoria cujo <label> em inglês seja 'e-Science Domain'.
- for \$c in (\$publicadoc//categories/category)
where \$c/label[@lang='en-US'] = 'e-Science Domain'
return \$c

count

- a) Retorne quantas publicações são posteriores ao ano de 2011.
- return
count(\$publicadoc // publication[year>2010])

Join

- Retorne as publicações associadas à categoria cujo <label> em inglês seja 'e-Science Domain'. A associação entre o label e a key da categoria deve ser feita na consulta.

```
for $c in ($publicadoc//categories/category),  
    $p in ($publicadoc//publication)  
where $c/label[@lang='en-US'] = 'e-Science  
Domain' and  
    $p/key = $c/@key  
return $p
```

Join

- Retorne as publicações associadas à categoria cujo <label> em inglês seja 'e-Science Domain'. A associação entre o label e a key da categoria deve ser feita na consulta.

```
for $c in ($publicadoc//categories/category),  
    $p in ($publicadoc//publication)  
where $c/label[@lang='en-US'] = 'e-Science  
Domain' and  
    $p/key = $c/@key  
return $p
```

```
LET $d := doc(www.company.com/info.xml)
FOR $x IN $d/company/project[projectNumber = 5]/projectWorker,
    $y IN $d/company/employee
WHERE $x/hours gt 20.0 AND $y.ssn = $x.ssn
RETURN <res> $y/employeeName/firstName, $y/employeeName/lastName,
    $x/hours </res>
```

-
1. FOR \$x IN
doc(www.company.com/info.xml)
//employee [employeeSalary gt 70000]/employeeName
RETURN <res> \$x/firstName, \$x/lastName </res>
 2. FOR \$x IN
doc(www.company.com/info.xml)/company/employee
WHERE \$x/employeeSalary gt 70000
RETURN <res> \$x/employeeName/firstName, \$x/employeeName/lastName </res>
 3. FOR \$x IN
doc(www.company.com/info.xml)/company/project[projectNumber = 5]/projectWorker,
\$y IN doc(www.company.com/info.xml)/company/employee
WHERE \$x/hours gt 20.0 AND \$y.ssn = \$x.ssn
RETURN <res> \$y/employeeName/firstName, \$y/employeeName/lastName, \$x/hours </res>

Figure 12.7

Some examples of XQuery queries on XML documents that follow the XML schema file *company* in Figure 12.5.

(Elmasri, 2011)

```
LET $d := doc(www.company.com/info.xml)
FOR $x IN $d/company/project[projectNumber = 5]/projectWorker,
    $y IN $d/company/employee
WHERE $x/hours gt 20.0 AND $y.ssn = $x.ssn
RETURN <res> $y/employeeName/firstName, $y/employeeName/lastName,
    $x/hours </res>
```

-
1. FOR \$x IN
doc(www.company.com/info.xml)
//employee [employeeSalary gt 70000]/employeeName
RETURN <res> \$x/firstName, \$x/lastName </res>
 2. FOR \$x IN
doc(www.company.com/info.xml)/company/employee
WHERE \$x/employeeSalary gt 70000
RETURN <res> \$x/employeeName/firstName, \$x/employeeName/lastName </res>
 3. FOR \$x IN
doc(www.company.com/info.xml)/company/project[projectNumber = 5]/projectWorker,
\$y IN doc(www.company.com/info.xml)/company/employee
WHERE \$x/hours gt 20.0 AND \$y.ssn = \$x.ssn
RETURN <res> \$y/employeeName/firstName, \$y/employeeName/lastName, \$x/hours </res>

Figure 12.7

Some examples of XQuery queries on XML documents that follow the XML schema file *company* in Figure 12.5.

(Elmasri, 2011)

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.

Referências Bibliográficas

- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados**. Pearson, 6a edição em português.
- Lee, T.B. **Notation 3**. March 2006. on-line:
<http://www.w3.org/DesignIssues/Notation3.html>
- Lee, T.B.; Hendler, J. & Lassila, O. **The Semantic Web**. Scientific American, 2001, 284, 28-37
- Moats R. **URN Syntax**. Network Working Group, May 1997.
- Sollins, K. and Masinter, L. **Functional Requirements for Uniform Resource Names**. Network Working Group, December 1994.
- Wang, X.; Gorkitsky, R. & Almeida, J.S. **From XML to RDF: how semantic web technologies will change the design of 'omic' standards** Nat Biotech. 2005, 23, 1099-1103.
- Whiteside, Arliss. **URNs of definitions in ogc namespace**. version: 1.0.0, document: 05-010. January 2005.

Referências Bibliográficas

- Leise, F.; Fast, K.; Steckel, M. **What Is A Controlled Vocabulary?** Boxes and Arrows, Dezembro 2002, online:
http://www.boxesandarrows.com/view/what_is_a_controlled_vocabulary_
- Amy J. Warner. **Taxonomy Primer**, online:
<http://www.lexonomy.com/publications/aTaxonomyPrimer.html>,
visitado em 20/08/2010.
- Wellisch, H. **Indexing from A to Z**. New York: H.W. Wilson, 1995. p. 214.
- Wilde, Erik. **XML Foundations** (slides). UC Berkeley iSchool, Aug 2006. <http://dret.net/lectures/xml-fall06/basics>

A black and white photograph of a spider web with several water droplets of varying sizes. The web is stretched across the frame, and the droplets are scattered across it, some larger and more prominent than others. The background is a soft, out-of-focus grey.

André Santanchè

<http://www.ic.unicamp.br/~santanche>

License

- These slides are shared under a Creative Commons License. Under the following conditions: Attribution, Noncommercial and Share Alike.
- See further details about this Creative Commons license at: <http://creativecommons.org/licenses/by-nc-sa/3.0/>