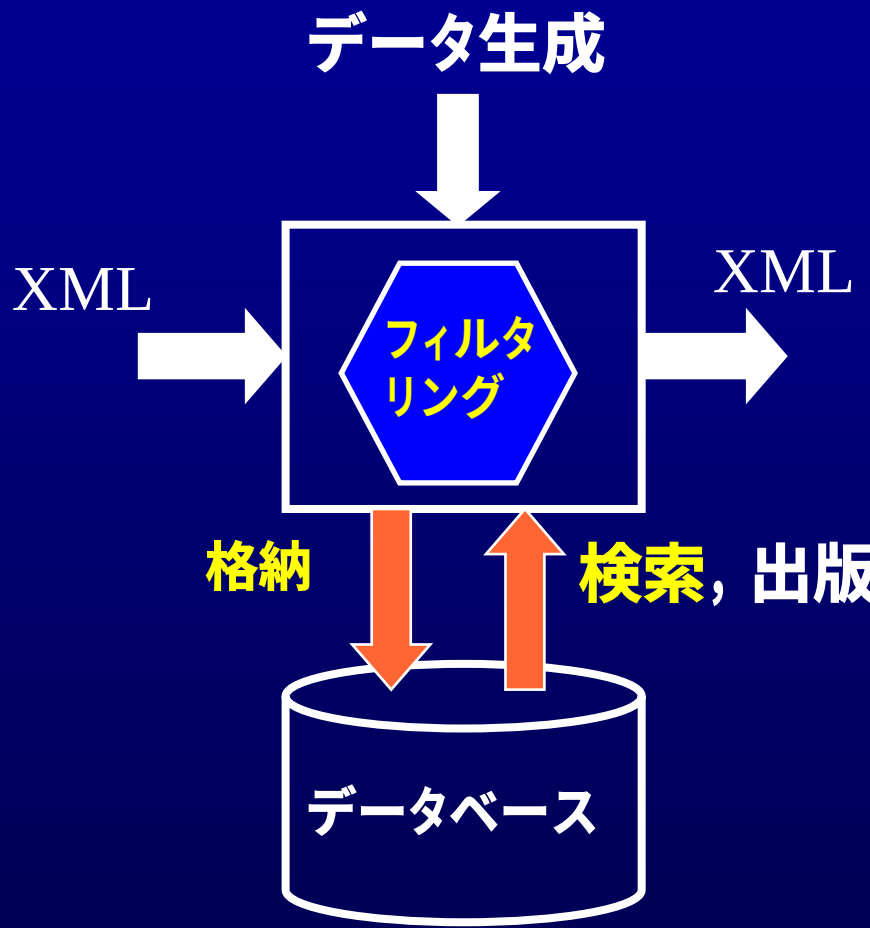
優れたWebDBの機能・性能要件と技術課題，応用について
--- データベースとXML ---

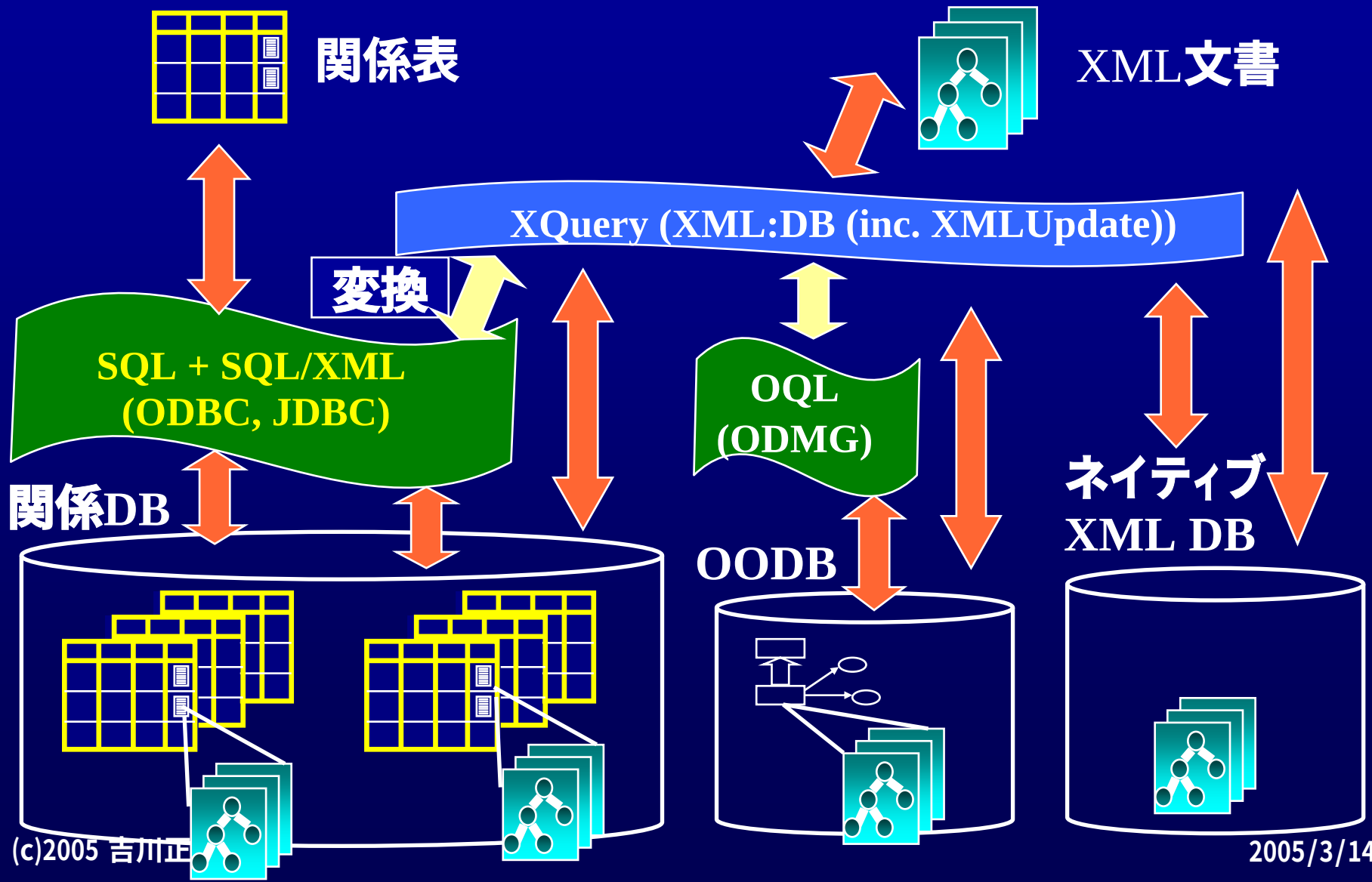
名古屋大学
情報連携基盤センター
吉川正俊

お話しすること/しないこと

- XML文書のデータベースへの格納
- XMLサーチエンジン
- XML索引
- 構造結合
- XML出版
- XMLストリーム処理



XML文書の格納，検索，出版



XPath 1.0

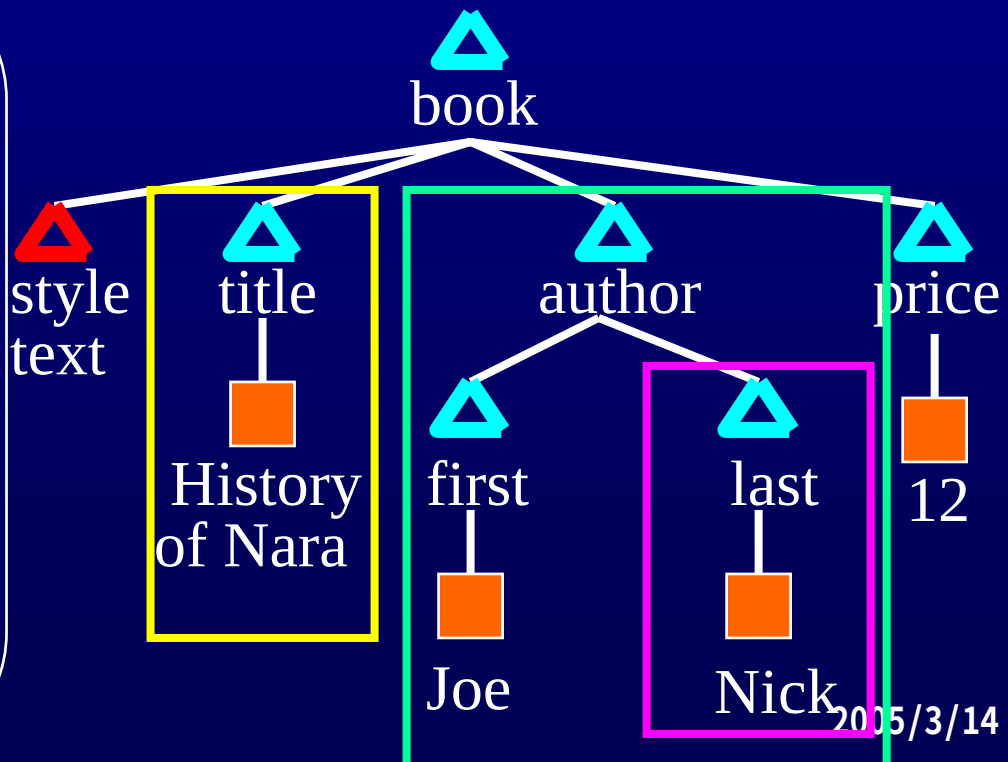
XPath1.0の例（その1）

/book/title

//last

//author[first='Joe']

```
<book style='text'>
 <title>History of Nara</title>
 <author>
   <first>Joe</first>
   <last>Nick</last>
 </author>
<price>12</price>
</book>
```



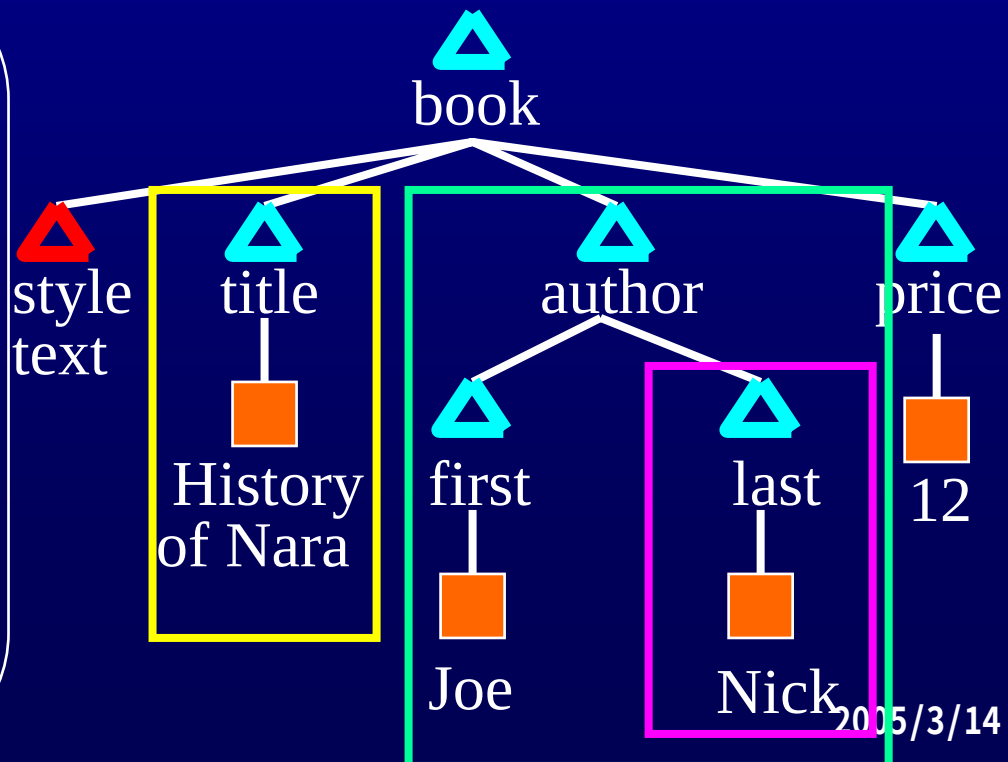
XPath1.0の例（その2）

/book[author/last='Nick']/title

//last[../first='Joe']

//author[first='Joe'][../price='12']

```
<book style='text'>
 <title>History of Nara</title>
 <author>
   <first>Joe</first>
   <last>Nick</last>
 </author>
<price>12</price>
</book>
```



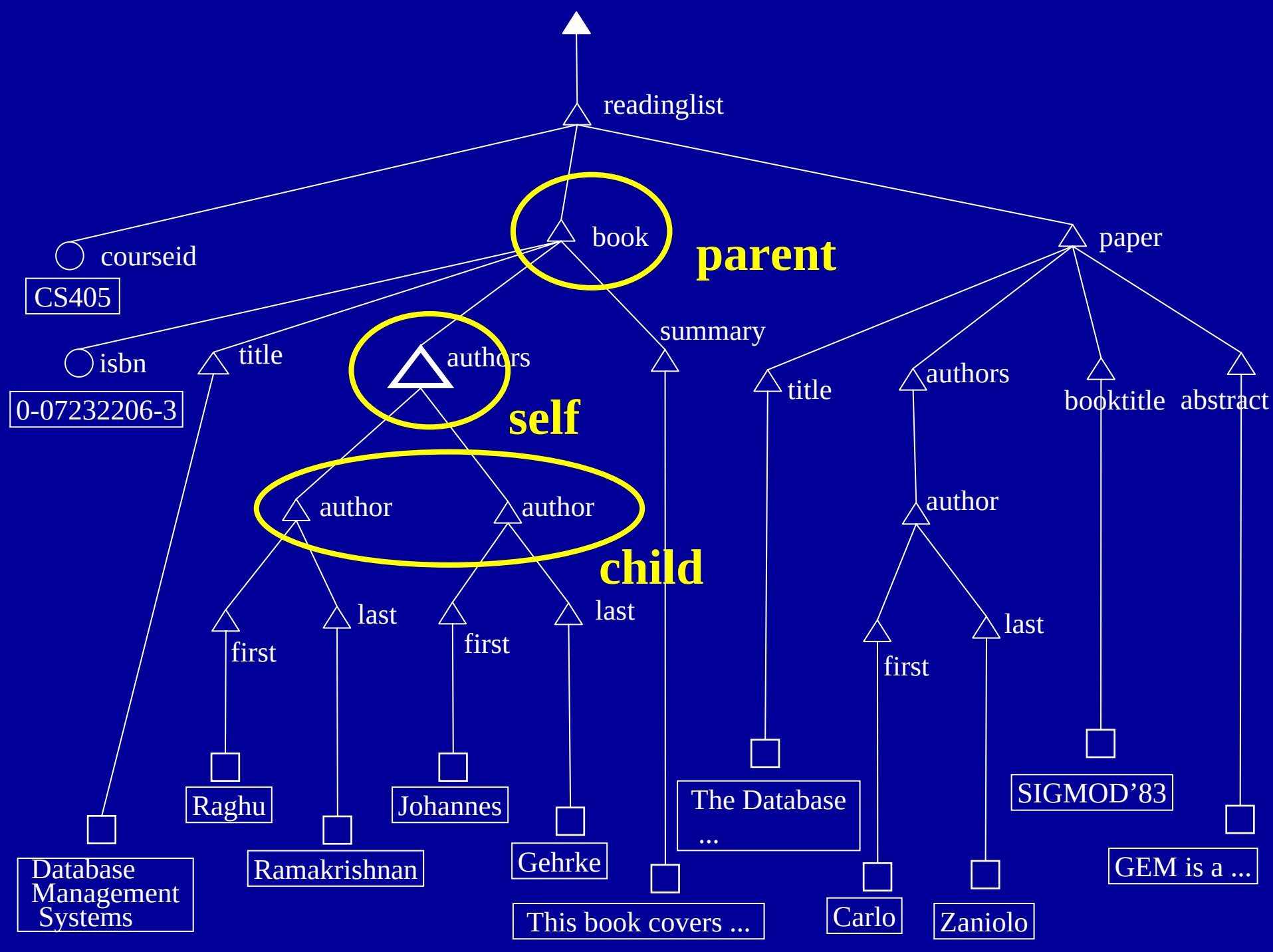
軸 (axis)

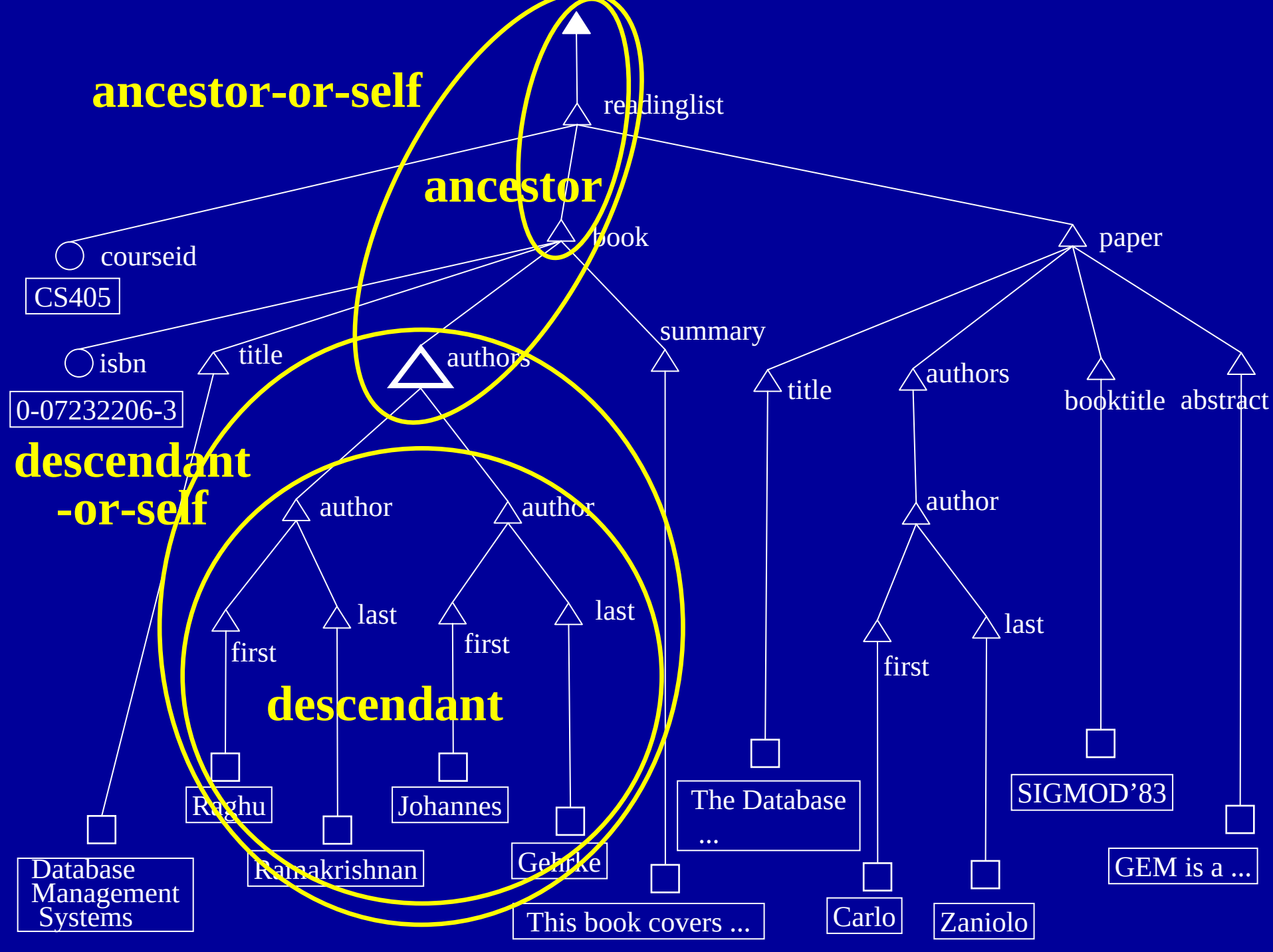
コンテクストノードからどちらの方向にノードを探しに行くか？

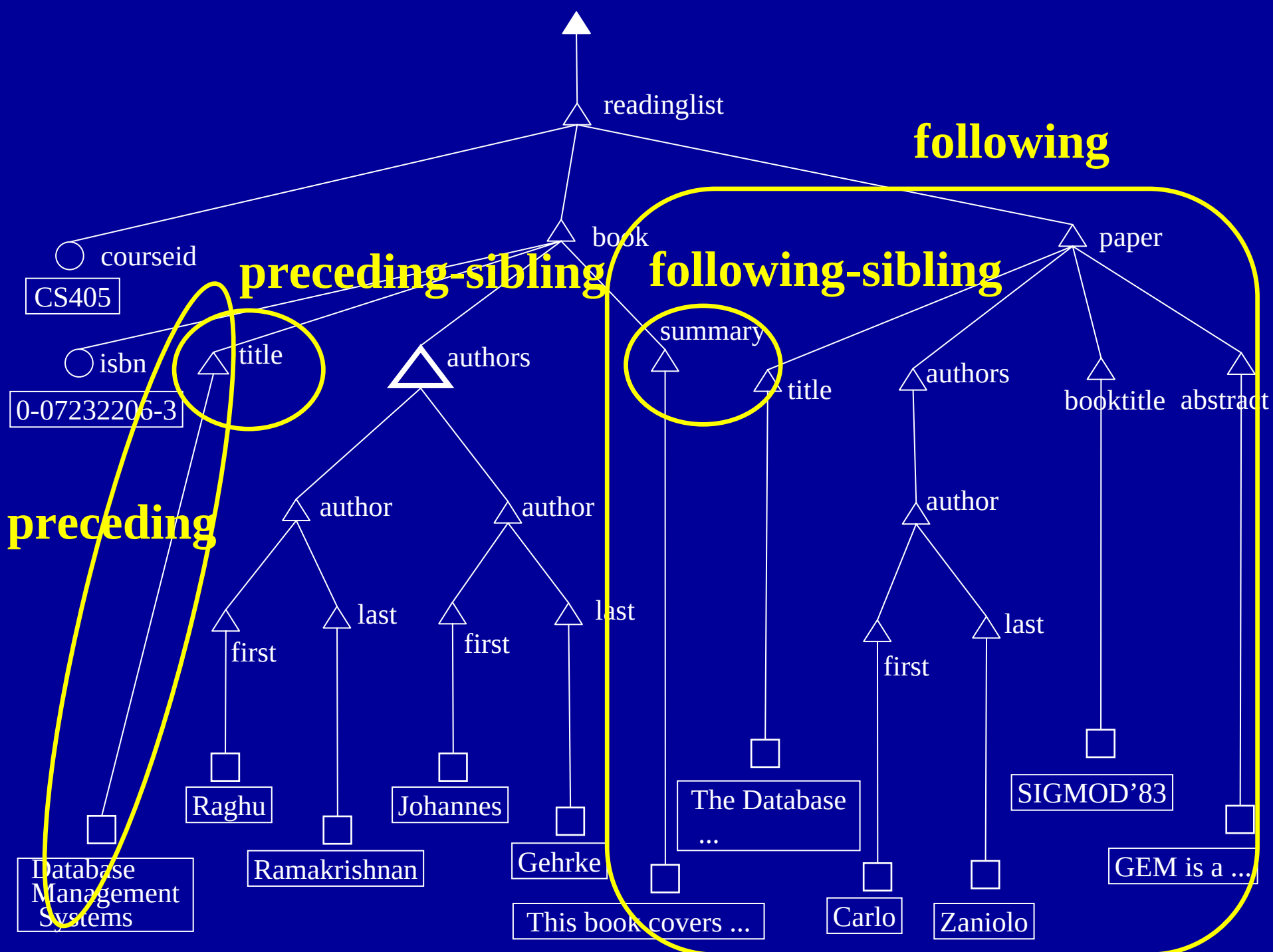
- ancestor
- ancestor-or-self
- parent
- self
- child
- descendant-or-self
- descendant
- attribute
- namespace
- following

  document orderでcontext nodeよりも後のノード．ただし，子孫や属性ノード，名前空間ノードは除く．
- following-sibling
- preceding

  document orderでcontext nodeよりも前のノード．ただし，先祖や属性ノード，名前空間ノードは除く．
- preceding-sibling


readinglist
book
parent
courseid
CS405
isbn
0-07232206-3
title
authors
self
summary
author
author
child
first
last
first
last
Raghu
Ramakrishnan
Johannes
Gehrke
Database Management Systems
This book covers ...
paper
title
authors
booktitle
abstract
author
first
last
The Database ...
Carlo
Zaniolo
SIGMOD'83
GEM is a ...


ancestor-or-self
ancestor
descendant
-or-self
descendant
readinglist
courseid
CS405
isbn
0-07232206-3
book
title
authors
summary
paper
title
authors
booktitle
abstract
author
author
author
first
last
first
last
first
last
Raghu
Ramakrishnan
Johannes
Gehrke
Database
Management
Systems
This book covers ...
The Database
...
Carlo
Zaniolo
SIGMOD’83
GEM is a ...


readinglist
following
courseid
CS405
book
paper
preceding-sibling
following-sibling
isbn
0-07232206-3
title
authors
summary
title
authors
booktitle
abstract
preceding
author
author
author
last
last
last
first
first
first
Raghu
Johannes
The Database
...
SIGMOD'83
Database
Management
Systems
Ramakrishnan
Gehrke
This book covers ...
Carlo
Zaniolo
GEM is a ...

XPath2.0 と XQuery1.0

	XPath1.0	XPath2.0	XQuery1.0
経路式		+範囲述語	- namespace軸はない
算術式, 比較式, 論理式			
組込み関数			
列式			
限量式			
条件式			
For式			FLWOR式
列型に関する式			型スイッチ式
妥当性検証式			
unordered式			
構築子			
問合せ前文			

XQuery1.0 --- FLWOR式

FOR
LET
WHERE
ORDER BY
RETURN

- 繰り返しや，変数を中間結果に束縛するために用いる．
- 二つ以上の文書の結合やデータの再構成に有用

XQuery1.0 --- グループ化

本または論文を3冊以上書いている著者

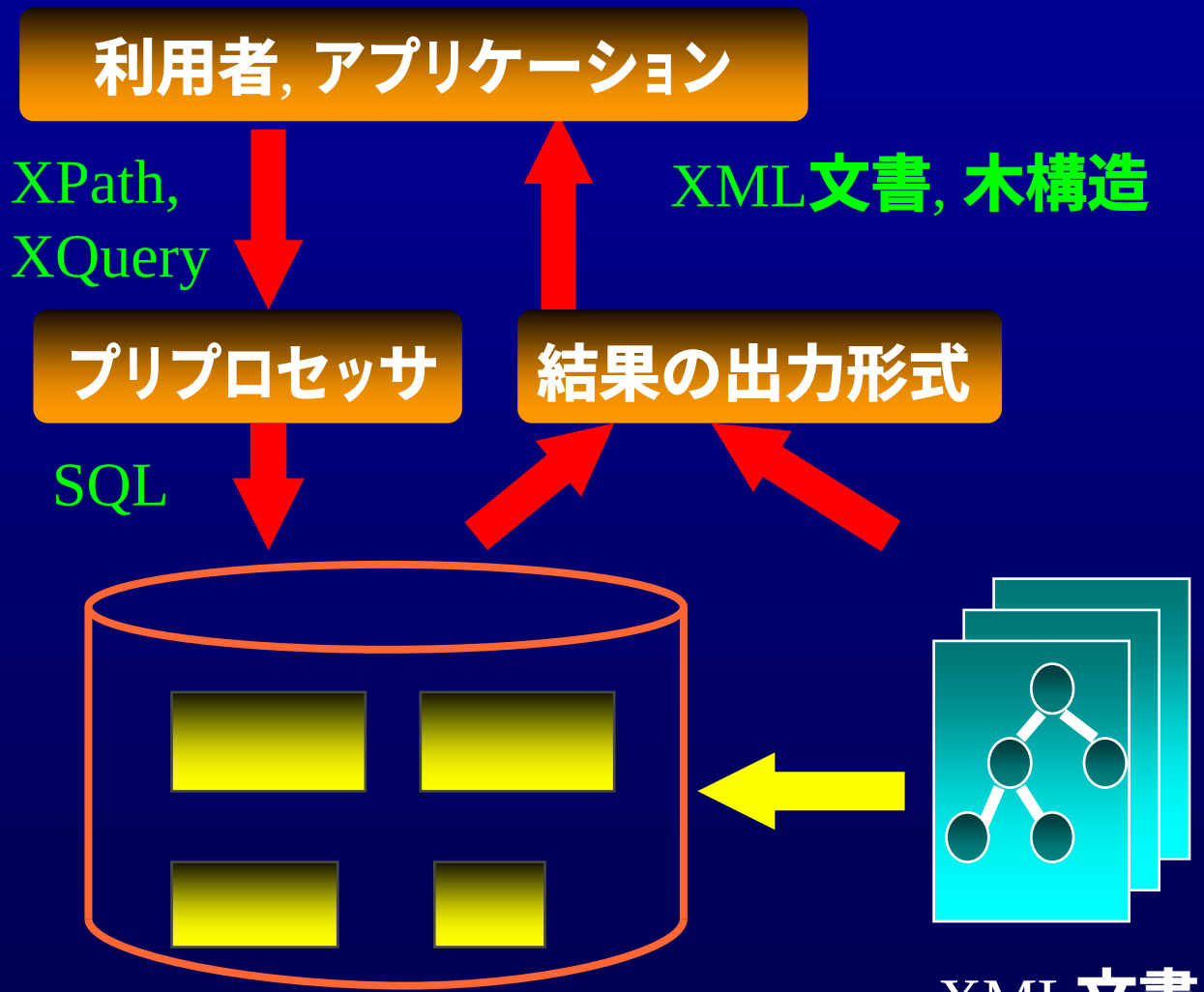
```
for $a in distinct-values(document("readinglist.xml")//author)
let $i := document("readinglist.xml")//*[authors/author = $a]
where count($i) >= 3
return
   <active-author>
         {
               $a,
               for $x in //*[authors/author = $a]
               return $x/title
         }
   </active-author>
```

XML文書のRDBへの格納

XML文書のRDBへの格納

考慮すべき点

- XMLスキーマの有無
- XPath, XQueryからSQLへの問合せ変換
- 問合せの処理効率
- データ容量
- 更新に対する頑健性



XML文書を分解し，既存DBMSのスキーマに写像した上で検索する方法

- 構造写像アプローチ
  - XML文書の論理構造（DTD）を表現するデータベーススキーマを定義
- モデル写像アプローチ
  - XMLデータモデルの構成要素（ノード，枝，経路など）を表現するデータベーススキーマを定義

×

- RDB
- OODB

XML文書を分解しデータベーススキーマに写像する方法 --- 関係データベースの場合

```
<emp>
        <name>Joe Doe</name>
        <age>30</age>
</emp>
```

構造写像アプローチ

モデル写像アプローチ

emp

name	age
Joe Doe	30

node

id	parent	name	value
1	-	emp	
2	1	name	Joe Doe
3	1	age	30

少数の文書構造 (DTD) に従う大量のXML文書

DTDが不明または頻繁に更新される

構造写像アプローチ

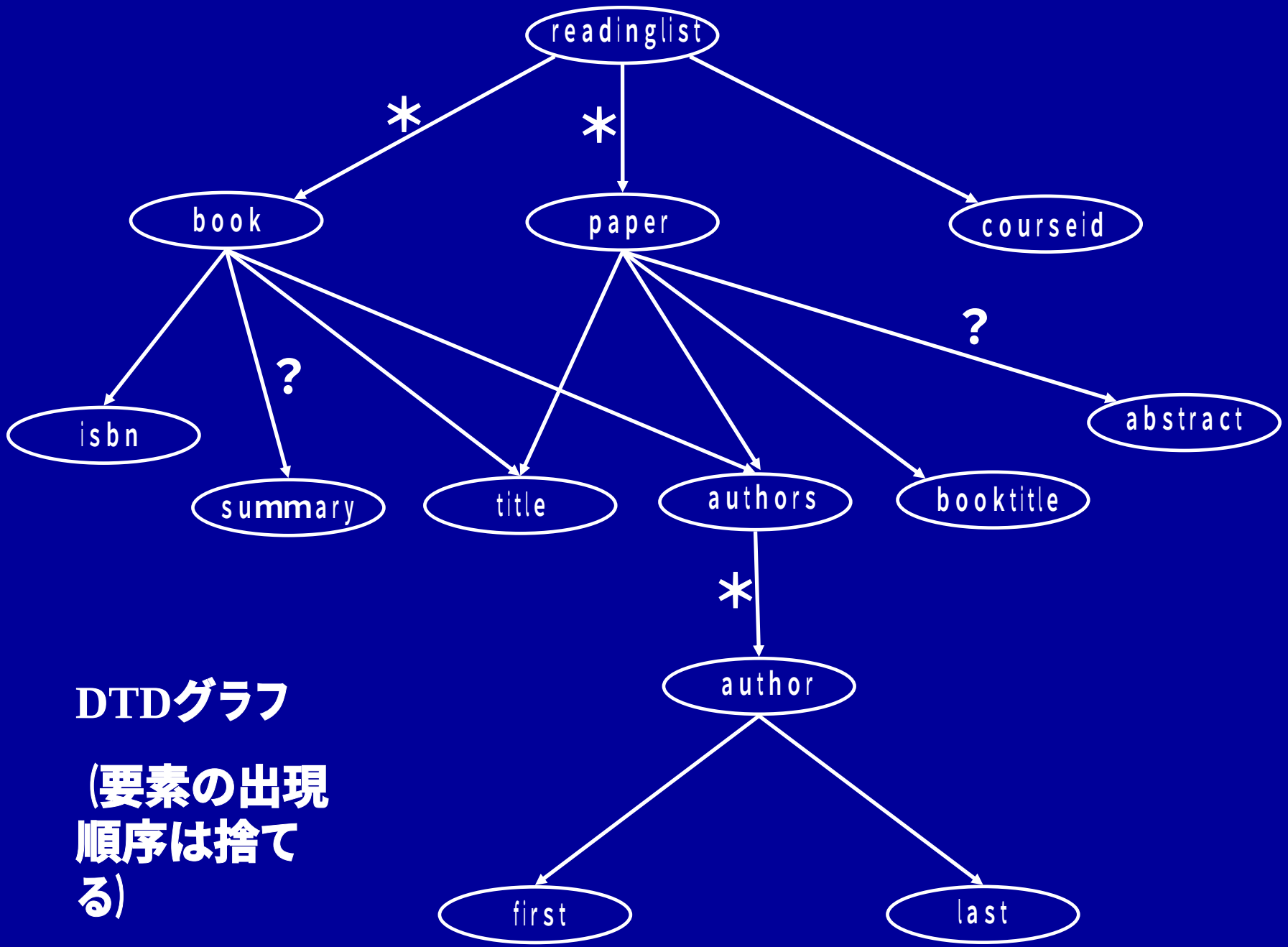
構造写像アプローチの例

Wisconsin Univ., VLDB'99

- DTDの構造を解析し，自動的に関係スキーマに変換
- QLは，XML-QLを使う
- XML-QLからSQLへの変換はshared approachに基づく
- DB2は recursionをsupportしているので使うことにした

DTDの例

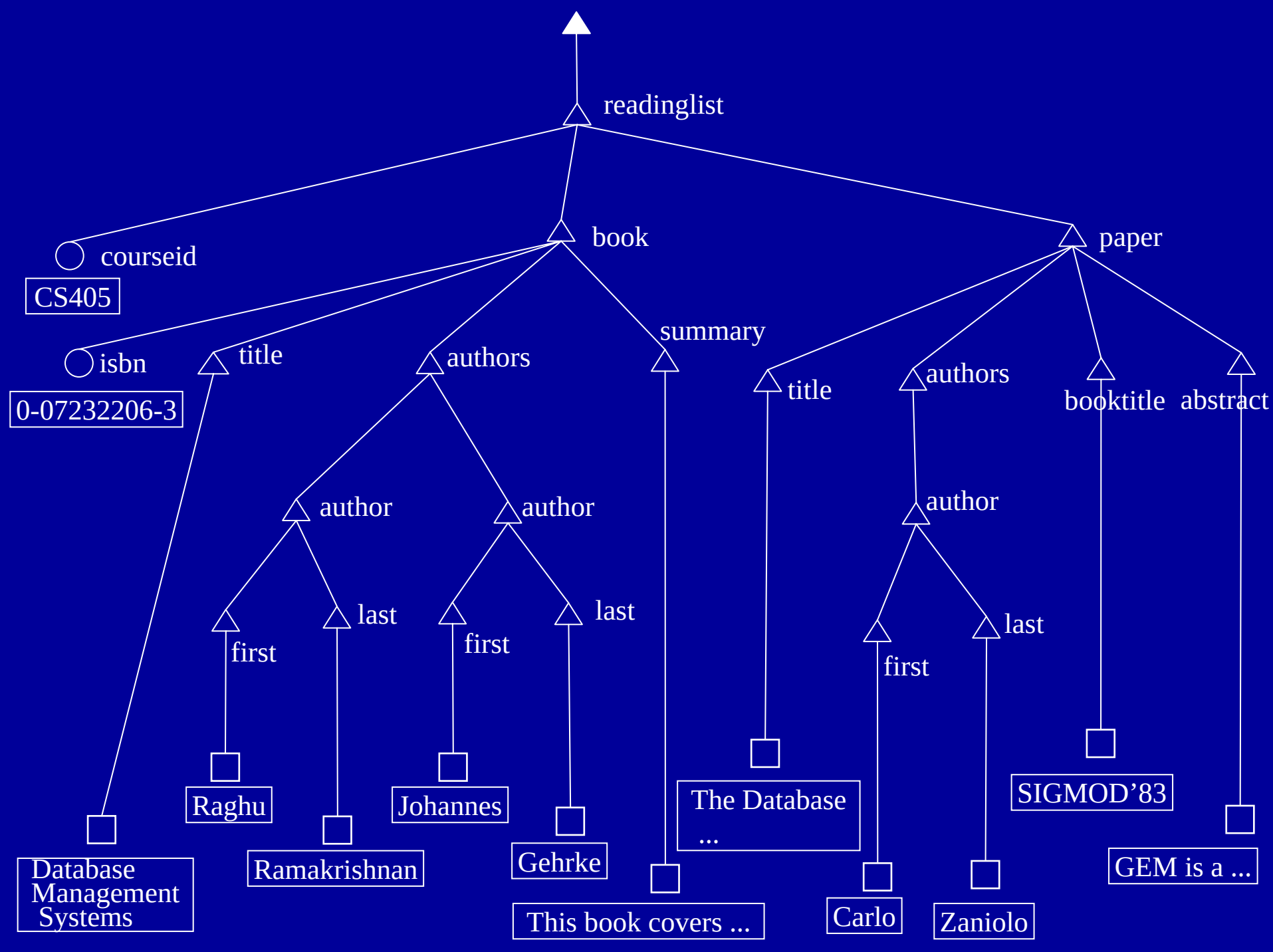
```
<!ELEMENT readinglist (book*, paper*)>
<!ATTLIST readinglist courseid ID REQUIRED>
<!ELEMENT book (title, authors, summary?)>
<!ATTLIST book isbn ID REQUIRED>
<!ELEMENT paper (title, authors, booktitle, abstract?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT authors (author+)>
<!ELEMENT author (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT summary (#PCDATA)>
<!ELEMENT booktitle (#PCDATA)>
<!ELEMENT abstract (#PCDATA)>
```



DTDグラフ

（要素の出現順序は捨てる）

```
<readinglist courseid="CS405">
   <book isbn="0-07-232206-3">
        <title>Database Management Systems</title>
        <authors>
                <author>
                        <first>Raghu</first>
                        <last>Ramakrishnan</last>
                </author>
                <author>
                        <first>Johannes</first>
                        <last>Gehrke</last>
                </author>
        </authors>
        <summary>This book covers the fundamentals of
                modern database management systems, ...
        </summary>
   </book>
   <paper>
        <title>The Database Language GEM</title>
        <authors>
                <author>
                        <first>Carlo</first>
                        <last>Zaniolo</last>
                </author>
        </authors>
        <booktitle>SIGMOD'83</booktitle>
        <abstract>GEM is a general-purpose query and
                update language for ...
        </abstract>
   </paper>
</readinglist>
```


readinglist
courseid
CS405
book
paper
isbn
0-07232206-3
title
authors
summary
title
authors
booktitle
abstract
author
author
author
first
last
first
last
first
last
Database Management Systems
Raghu
Ramakrishnan
Johannes
Gehrke
This book covers ...
The Database ...
Carlo
Zaniolo
SIGMOD'83
GEM is a ...

DTDグラフを関係スキーマに変換する主要な方法

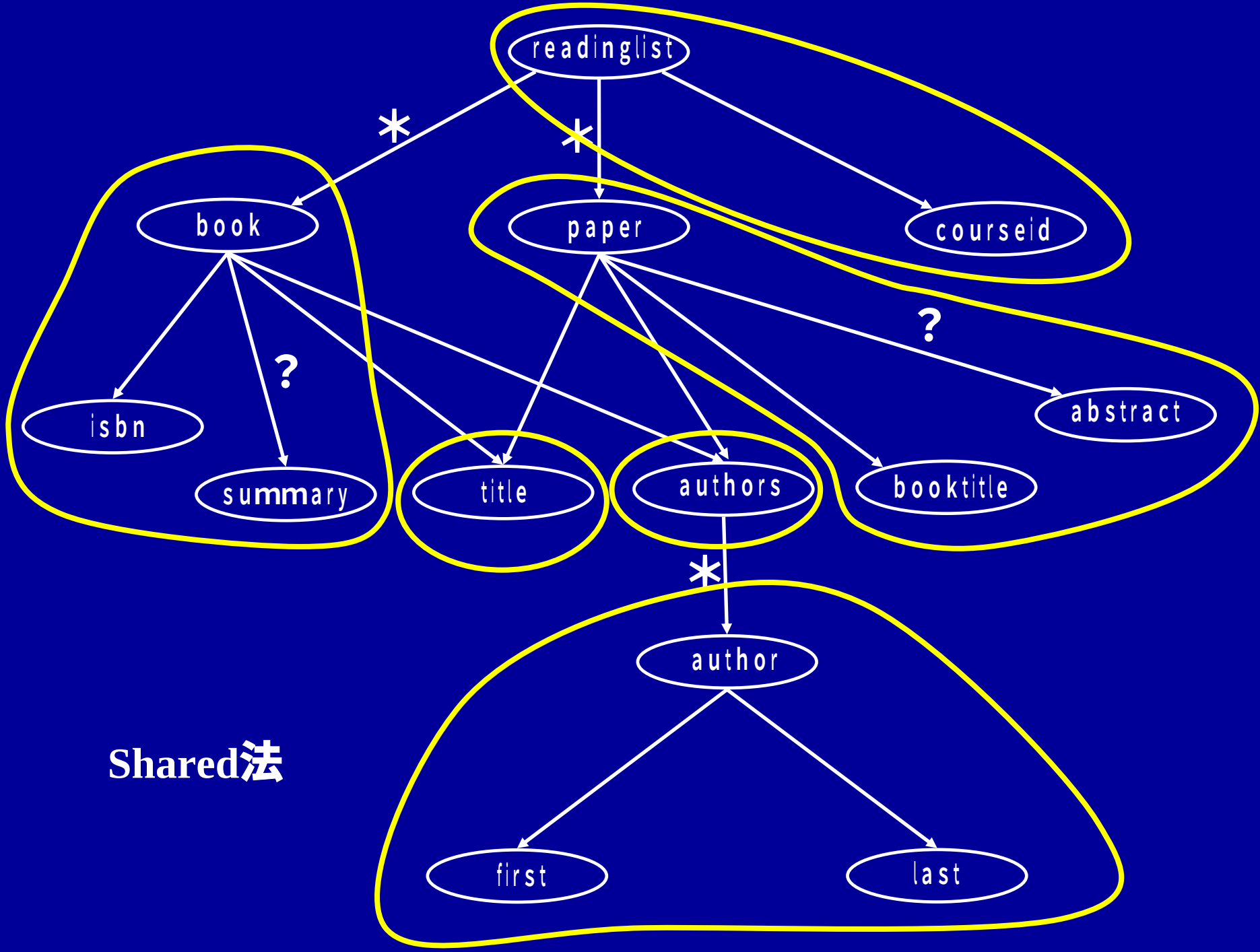
Shared法

DTDグラフにおいて入次数が2以上の要素（すなわち複数個の要素に共有されている要素）に対応して関係スキーマを作る．入次数が1の要素はその先祖の要素に対応する関係の属性に変換する．また，入次数が0の要素も対応する関係スキーマを作る．ただし，``*"の下の要素は別の関係スキーマとする．これは，関係データベースでは集合値をそのまま格納できないためである．その上で，関係スキーマ（Rとする）が作られる要素からDTDグラフ内の有向経路に沿って到達できる他の要素も関係スキーマRにインライン化する（すなわち，関係スキーマRの属性とする）．ただし，この有向経路には``*"を含んではならない．

Hybrid法

入次数が2以上の要素でも他の要素から``*"を通らずに到達可能な要素はインライン化する．

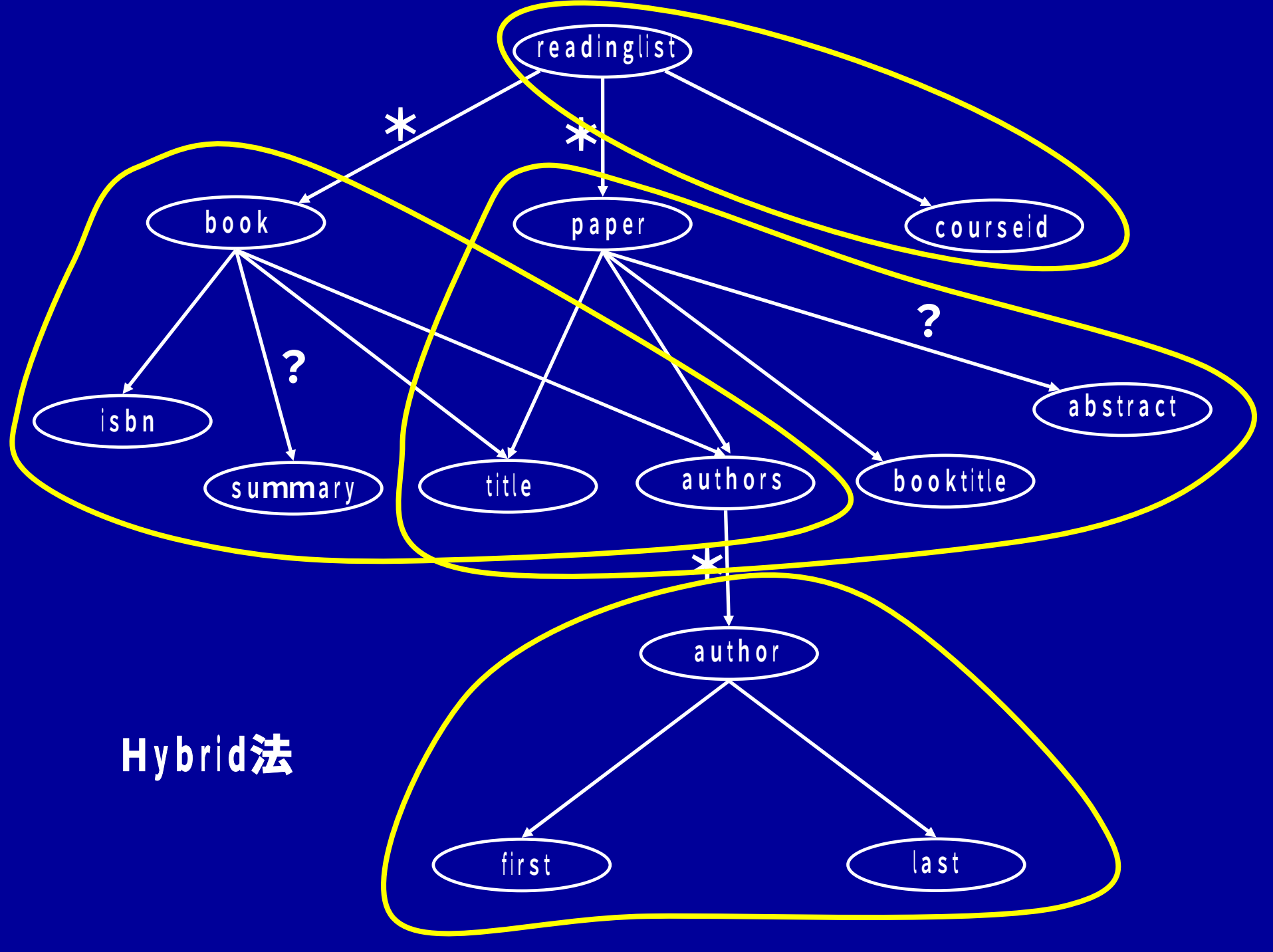
なお，最初のDTD簡単化において捨てられた要素の出現順序情報は，関係スキーマ中に要素の出現位置を追加することによって保持できる．


readinglist
*
*
book
paper
courseid
?
?
isbn
summary
title
authors
booktitle
abstract
*
author
first
last

Shared法

関係スキーマ (shared法)

readinglist(readinglistID: integer,
readinglist.courseid: string)
book(bookID: ingeger, book.parentID: integer,
book.isbn: string, book.summary, string)
paper(paperID: integer, paper.parentID: integer,
paper.booktitle: string, paper.abstract: string)
title(titleID: integer, title.parentID: integer,
title.parentCODE: integer, title: string)
authors(authorsID: integer, authors.parentID: integer,
authors.parentCODE: integer)
author(authorID: integer, author.parentID: integer,
first: string, last: string)


readinglist
*
*
book
paper
courseid
?
?
isbn
summary
title
authors
booktitle
abstract
*
author
first
last
Hybrid法

関係スキーマ (hybrid法)

readinglist(readinglistID: integer,
readinglist.courseid: string)
book(bookID: ingeger, book.parentID: integer,
book.isbn: string, book.summary: string,
book.title: string, book.authors.authorsID: integer)
paper(paperID: integer, paper.parentID: integer,
paper.booktitle: string, paper.abstract: string,
paper.title:string, paper.authors.authorsID: integer)
author(authorID: integer, author.parentID: integer,
author.parentCODE: integer, first: string,
last: string)

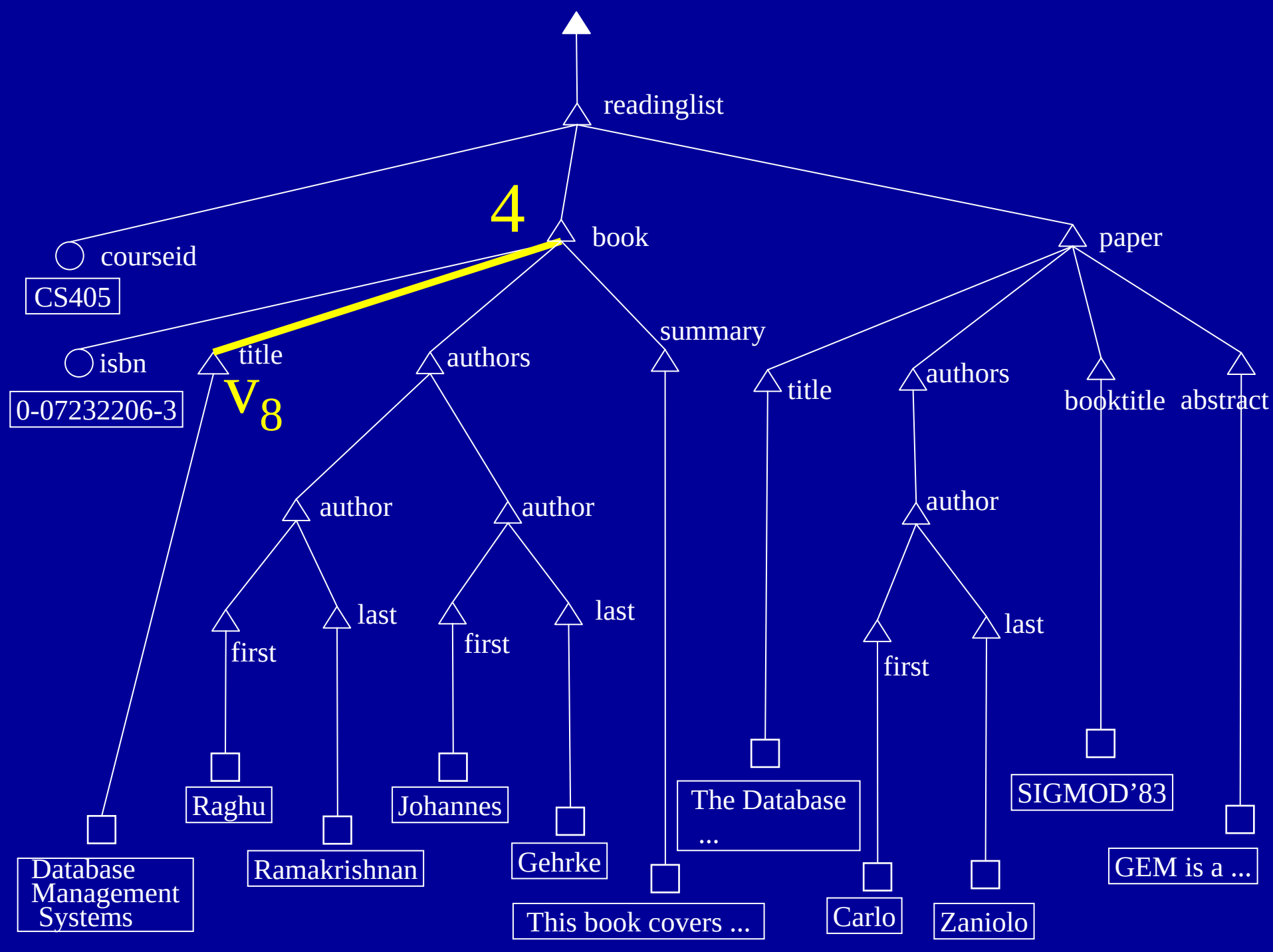
モデル写像アプローチ

モデルアプローチに基づくRDBへの格納

- XMLスキーマの存在とは独立
  - どのような整形式 XML文書も格納可能
- 問題
  - どのような関係データベーススキーマを設計するか？
    - » 枝アプローチ
    - » 経路アプローチ
  - XML問合せからSQLへの変換をどのように行うか？

枝アプローチ

- XML木の各枝を関係の組として格納する.
- 各ノードに唯一の識別子が付与されていることを前提とする


readinglist
4
courseid
CS405
book
paper
isbn
0-07232206-3
title
v_8
authors
summary
title
authors
booktitle
abstract
author
author
author
last
last
last
first
first
first
Raghu
Johannes
The Database ...
SIGMOD'83
Database Management Systems
Ramakrishnan
Gehrke
This book covers ...
Carlo
Zaniolo
GEM is a ...

枝アプローチ

- 問合せ変換
  - 経路式の長さに比例した結合が必要

XPath: /readinglist/book/title

SQL:
```
SELECT e3.target
FROM Edge e1, Edge e2, Edge e3
WHERE e1.name = “readinglist”
AND e2.name = “book”
AND e3.name = “title”
AND e1.target = e2.source
AND e2.target = e3.source
```

経路アプローチ

経路アプローチ

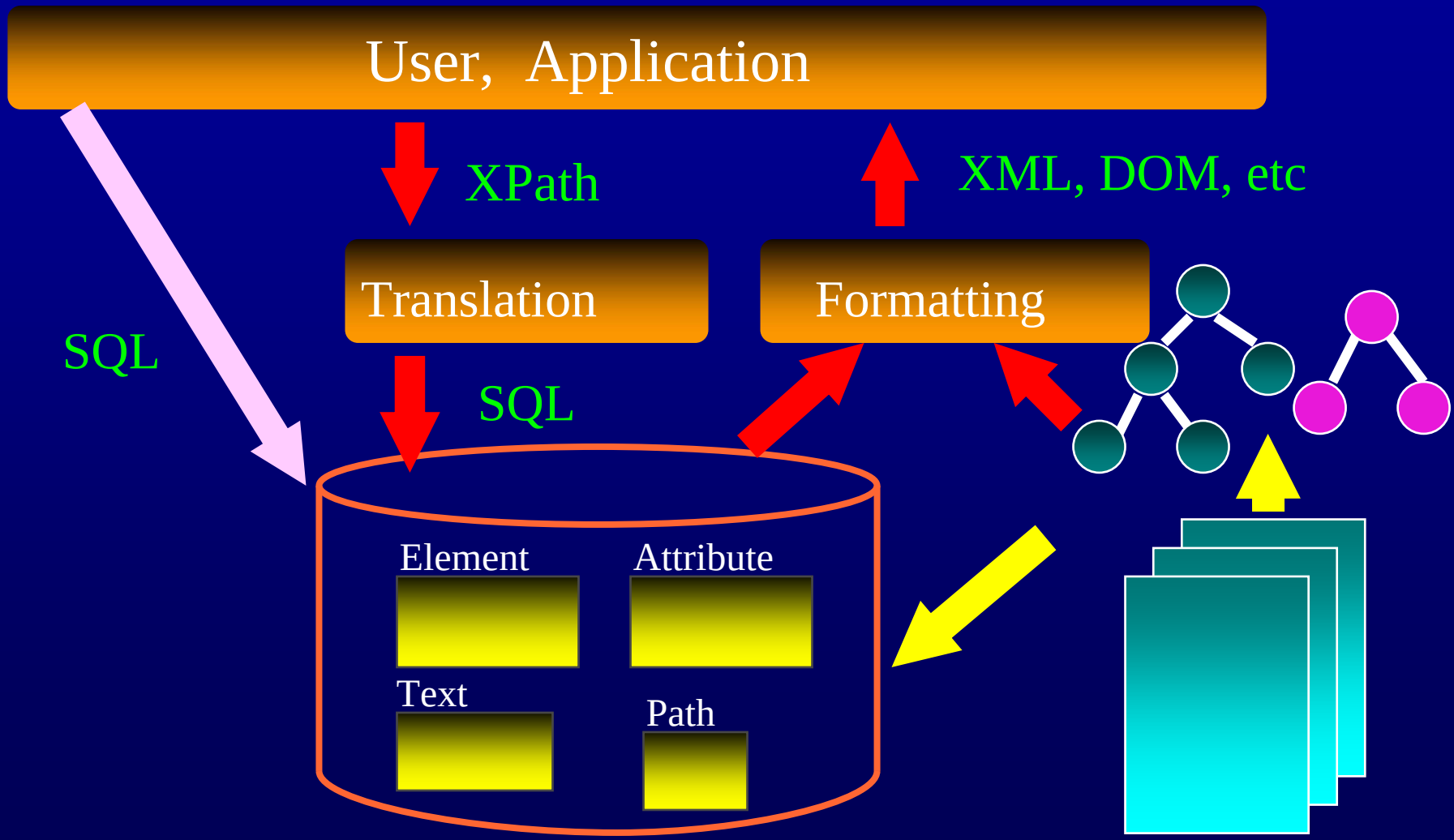
- XML木の経路を格納単位とする
- ノードのラベル付け
  - XML木のトポロジーの保存
  - 問合せ，更新処理の高速化
- XPathで表される経路に関する複雑な条件
  - SQLの文字列パターン照合と整数の比較に変換
    - » 複雑な結合処理は不要
  - 通常の DBMS で提供されている索引を利用可 (B+木，R木)

木のノードラベリング手法

- 考慮すべき点
  - XPathの軸（特に親子，先祖/子孫関係）の高速計算
  - 小容量
  - XMLスキーマの考慮
  - 更新に対する頑健性

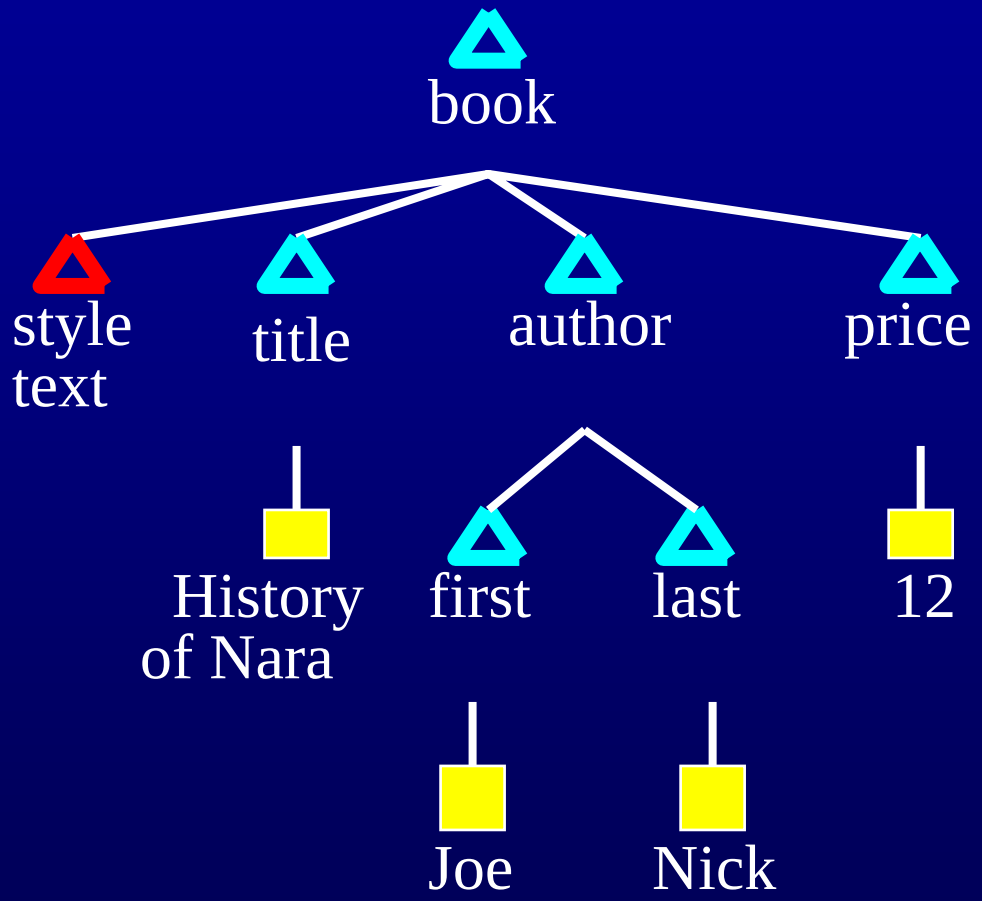
XRel

Storing XML Documents in RDB



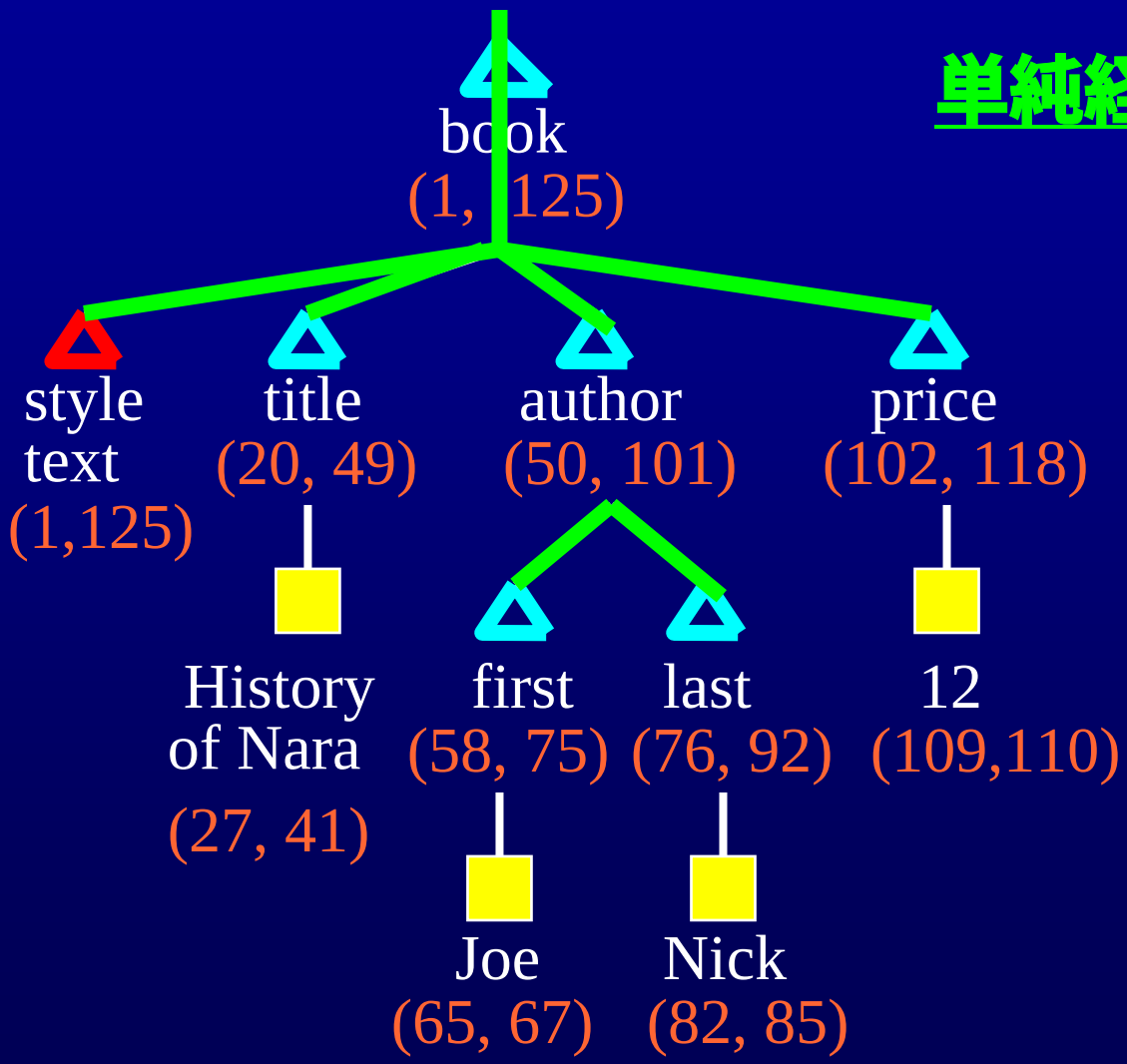
XML文書

```
<book style='text'>
  <title>History of Nara</title>
  <author>
    <first>Joe</first>
    <last>Nick</last>
  </author>
  <price>12</price>
</book>
```



Element Attribute Text

XML文書の分解，表現の単位



単純経路式

/book

/book/@style

/book/title

/book/author

/book/author/first

/book/author/last

/book/price

リージョン : (Start, End)

XML文書の関係への格納方法

Path

pathname	pathID
/book	1
/book/title	2
/book/author	3
…	•

Element

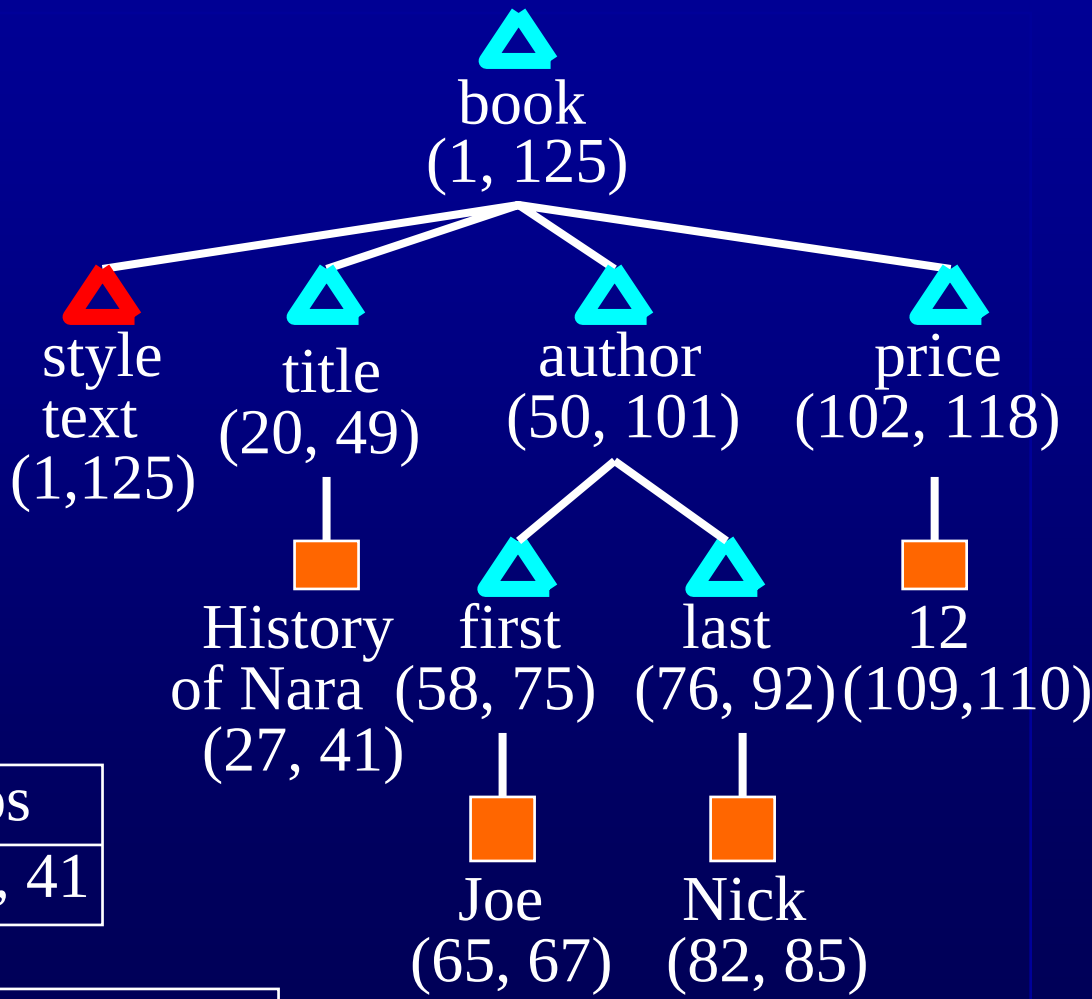
docID	pathID	pos
#1	2	20, 49

Text

docID	pathID	value	pos
#1	2	History	27, 41

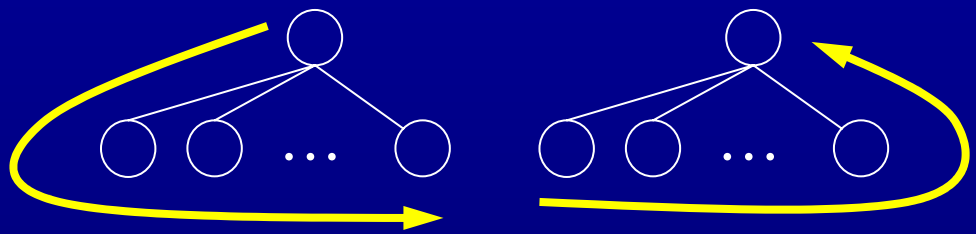
Attribute

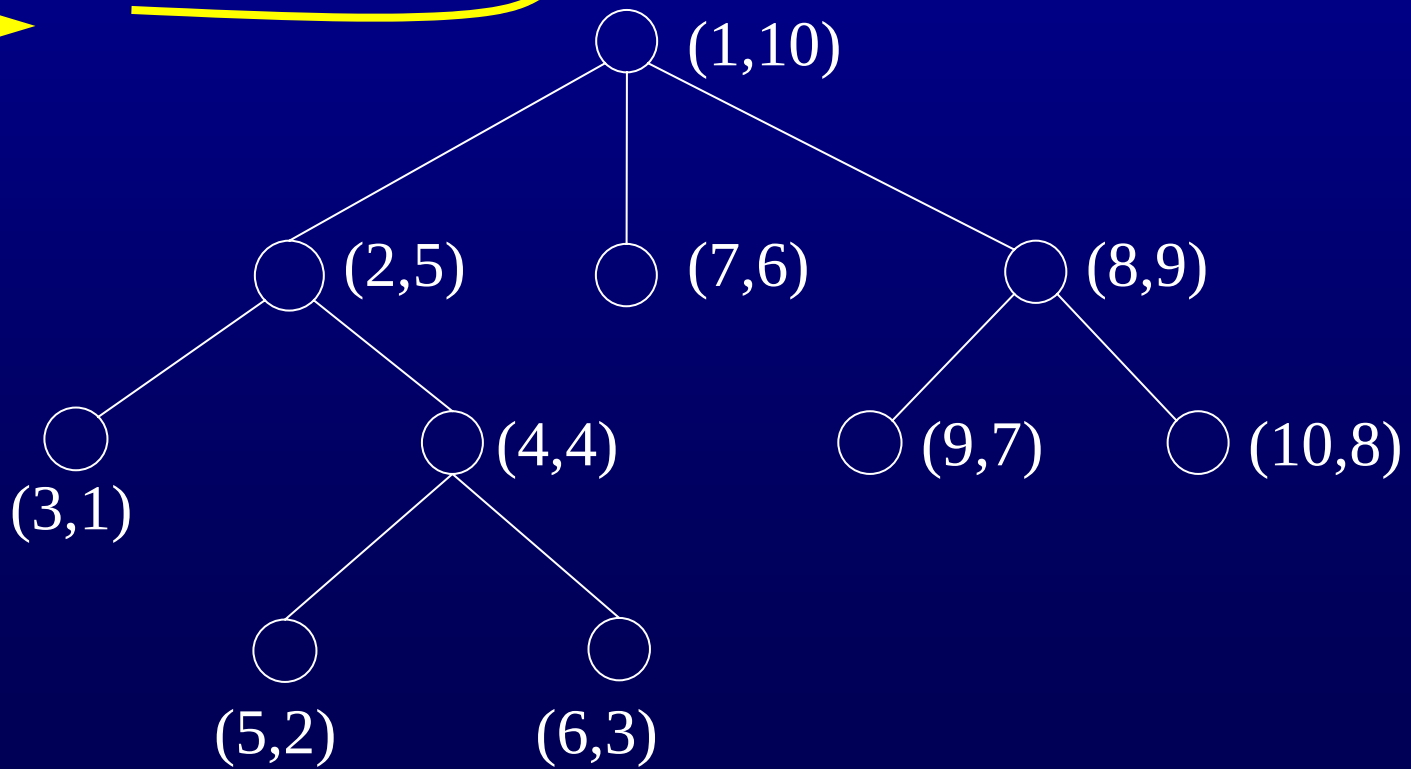
docID	pathID	attname	attval	pos
#1	1	style	text	1, 125



Dietzの番号付けスキーム

- (preorder, postorder)





Dietzの番号付けスキーム

- 順序関係は，リージョン（開始タグの先頭からのバイト数と終了タグの先頭からのバイト数）と同じ
- ノード a はノード d の先祖

  ⇔

  a.preorder < d.preorder　and　a.postorder > d.postorder

格納方法 ~ 索引の利用 ~

Element　B+木　R木

docID	pathID	pos
#1	1	1, 125
#1	2	20, 49
#1	3	50, 101
•	•	•••

Text　B+木　R木

docID	pathID	value	pos
#1	2	History.	27, 41
#1	4	Joe	65, 67
…	…	…	…

Attribute　B+木　R木

docID	pathID	attname	attval,	pos
#1	1	style	text	1, 125

Path　B+木

pathname	pathID
/book	1
/book/title	2
/book/author	3
…	•

検索方法~問合せの書き換え~

/book/title

//last

//author[first='Joe']

Path

pathname	pathID
/book	1
/book/title	2
/book/author	3
…	•

- XPathの基本は / または // と要素型をつないだ列
- データベースでは XML 文書を単なる文字列として格納
- 書き換えについては, SQL-92 のもつ文字列照合を行う述語を利用

問合せ変換例(1) ~単純経路式~

/book/title

```
SELECT e.docID, e.position
FROM   Element e, Path p
WHERE e.pathID = p.pathID
AND    p.pathname = "/book/title"
```

Element

docID	pathID	pos
#1	1	1, 125
#1	2	20, 49
#1	3	50, 101
•	•	•••

Path

pathname	pathID
/book	1
/book/title	2
/book/author	3
…	•

問合せ変換例(2) ~あいまい経路式~

//last

```
SELECT e.docID, e.position
FROM   Element e, Path p
WHERE e.pathID = p.pathID
AND    p.pathname like "%/last"
```

Element

docID	pathID	pos
#1	1	1, 125
#1	2	20, 49
...	...	...
#1	5	76, 92

Path

pathname	pathID
/book	1
/book/title	2
...	...
/book/author/last	5

文字列照合を行う述語

問合せ変換例(3)
~制約条件付きのあいまい経路式~

```
//author[first='Joe']

SELECT e.docID, e.position
FROM   Element e, Text t,
       Path p1, Path p2
WHERE  p1.pathname
         like "%/author"
AND    p2.pathname
         like "%/author/first"
AND    t.value = "Joe"
AND    t.pathID  = p2.pathID
AND    e.pathID = p1.pathID
AND    e.docID = t.docID
AND    e.pos.contain(t.pos)
```

Element

docID	pathID	pos
#1	1	1, 125
#1	2	20, 49
#1	3	50, 101

Text

docID	pathID	value	pos
#1	4	Joe	65, 67

Path

pathname	pathID
/book/author	3
/book/author/first	4

問題点

- Θ結合多用による性能劣化

XParent

XML文書例

```
<BOOK ISBN="1-55860-438-3">
        <SECTION>
                <TITLE>Bad Bugs</TITLE>
                Nobody loves bad bugs.
                <FIGURE CAPTION="Sample bug"/>
        </SECTION>
        <SECTION>
                <TITLE>Tree Frogs</TITLE>
                All right-thinking people
                <BOLD> love </BOLD> tree frogs.
        </SECTION>
</BOOK>
```

XParentのスキーマ

LabelPath

ID	Len	Path
1	1	/Book
2	2	/Book/@ISBN
3	2	/Book/Section
4	3	/Book/Section/Title
	…	

Data

PathID	Did	Ordinal	Value
2	&5	1	1-55860-438-3
4	&13	1	Bad Bugs
	…		

Element

PathID	Ordinal	Did
1	1	&1
2	1	&2
3	1	&3
3	2	&4
	…	

Data-Path
（ノード間の親子関係）

Pid	Cid
&1	&2
&1	&3
&1	&4
&2	&5
&3	&6
&3	&7
…	

Ancestor
（ノード間の先祖/子孫関係）

Did	Ancestor	Level
&2	&1	1
&3	&1	1
&4	&1	1
&5	&2	1
&5	&1	2
&6	&3	1
&6	&1	2
&7	&3	1
&7	&1	2
	…	

XParent

- 利点
  - Θ結合は不要．すべて等結合
- 問題点
  - データ容量が大きい．
  - 等結合ではあるが，多くの表の結合必要．特に，サイズが大きいAncestor表との結合は問題．

SUCXENT

SUCXENTのスキーマ

Path

ID	Path
1	/Book
2	/Book/@ISBN
3	/Book/Section
4	/Book/Section/Title
…	

AncestorInfo

(葉ノードの先祖情報)

Sibling Order	Ancestor Order (先祖のpreorder)	Ancestor Level (先祖の深さ)
#1	1	1
#2	1	1
#2	3	2
	…	

PathValue (葉ノードの情報を格納)

PathId	LeafOrder (葉の出現順)	SiblingOrder (兄弟グループ(出現順))	BranchOrder (直前のSiblinOrderの葉ノードとの共通祖先のレベル)	LeafValue
2	&1	#1	-1	1-55860-438-3
4	&2	#2	1	Bad Bugs
	…			

LargeText (長いテキストデータは，こちらに格納)

PathId	LeafOrder	SiblingOrder	BranchOrder	LeafValue
		…		

SUCXENT

- 問合せは候補の葉ノードを求め，それら葉ノードの最小共通祖先を求めれば答えることができる，という事実に着目
- 利点
  - Θ結合は不要
  - XParentとは異なり，葉ノードの先祖だけを格納
- 欠点
  - 葉ノードの先祖情報のみでも容量は大きい

Microsoft SQL Server 2005

問合せの変換

/BOOK[@ISBN = “1-55860-438-3”]/SECTION

引数のノードを根とする
部分文書を返す

```
SELECT SerializeXML (N2.ID, N2.ORDPATH)
FROM infosettab N1
        JOIN infosettab N2 ON (N1.ID = N2.ID)
WHERE N1.PATH_ID = PATH_ID(/BOOK/@ISBN)
AND N1.VALUE = '1-55860-438-3'
AND N2.PATH_ID = PATH_ID(BOOK/SECTION)
AND Parent (N1.ORDPATH) = Parent (N2.ORDPATH)
```

引数のORDPATHの親ノードのORDPATHを返す

引数の経路の
経路IDを返す

お話しすること/しないこと

- XML文書のデータベースへの格納
- XMLサーチエンジン
- XML索引
- 構造結合 (structural join)
- XML出版
- XMLストリーム処理

情報検索技術との融合

- 末端利用者は，XPathやXQueryを使いこなせるか？ ➡ No!!

- XMLサーチエンジンが必要
  - 最も単純な問合せは，キーワード集合
  - 検索対象の粒度は部分文書

課題

- 問合せ結果部分文書の判定
- 全文検索，近接検索

INEX Project (1/2)

- XML文書のための巨大テストコレクションを作成するプロジェクト
- 2002年発足
- 世界42グループが参加(2003年)
- プロジェクトリーダー
  - N. Fuhr (Germany), M. Lalmas (UK)
- DELOS Network of Excellence for Digital Libraries

INEX Project (2/2)

- テストコレクション
  - IEEE Computer Society Transactions/Magazines の 8年分 (1995-2002)，12,000件以上の論文
  - 対象XML文書サイズ 約500MB
  - 各論文は平均 1,500 XML ノードから成る
- 問合せ
  - content-only (CO)
    TREC同様，フリーテキスト問合せ
    検索システムは適切な粒度の部分文書を検索
  - content-and-structure (CAS) queries
    検索対象文書の構造を明示的に指定
  - 問い合わせのトピックの内容の広さは様々

今後の課題

- XQueryの処理高速化
- XMLスキーマの意識，利用
  → 自律性（スキーマ，索引の自動選択）
- XMLのための物理データベーススキーマ

- 自然言語入力
- 高度なXMLサーチエンジン
  - 情報検索技術との統合
- 細粒度の並行処理

- 時制データ，バージョンの扱い
- 時刻認証，個人認証
- アクセス権制御
- プライバシー保護
- 電子透かし