Обработка и исполнение запросов в СУБД (Лекция 9)

XML СУБД

v2

Георгий Чернышев

Академический Университет

chernishev@gmail.com

23 ноября 2016 г.

XML I

- XML — eXtensible Markup Language;
- Бесплатное упрощение SGML, первая версия в 1998, W3C;
- Идеи:
  1. добавим к тексту теги, для возможности придания семантики;
  2. дать возможность пользователю задавать свои теги;
  3. текст должен быть читабельным пользователем;
  4. возможность указания обработки;
- Пришел не один, а с “обвесом” других языков и технологий;
- Формирует базис многих технологий: ODF, XHTML, OOXML, DocBook.

XML II

Успешные приложения:

- Собственно, разметка документов;
- Обмен данными в слабо связанных системах:
  - Суть: DTD или XML Schema задает формат сообщений при общении различных компьютерных систем;
  - Не только описывает формат, но и структуру и данные сообщений;
  - Примеры: SOAP, RSS/Atom.
- Моделирование слабо-структурированных данных:
  - Суть: менее строгая нежели реляционная, ER и OO модели;
  - Иерархическая, гибкая, позволяет разреженность в данных;
  - Позволяет иметь гетерогенную структуру данных;
  - Позволяет иметь свои свойства для каждого объекта;
  - Способна к быстрым изменениям.

XML III

- Элемент (открывающий и закрывающий тег + тело):

```
<t1> body </t1>
```

- Элементы могут быть вложенными:

```
<t1> body1 <t2> body2 </t2> </t1>
```

- Тег может иметь атрибуты:

```
<t1 attr1=``value1''> body </t1>
```

- Элементы могут быть пустыми:

```
<t1/>
```

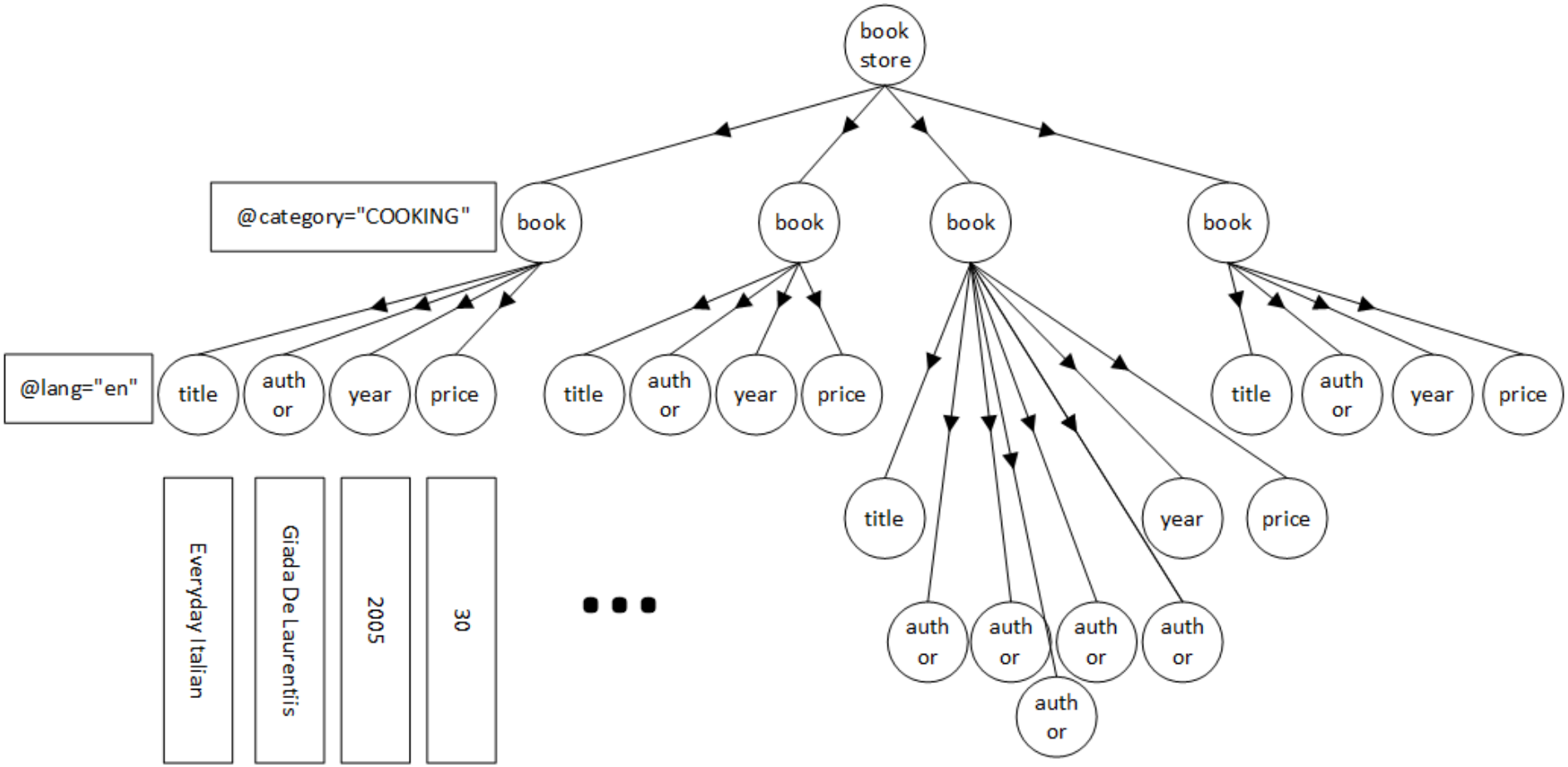
- DTD и XML schema ограничивают возможные деревья;
- В дереве есть порядок, левосторонний обход в глубину.

Пример I

Пример взят из
http://www.w3schools.com/xml/xquery_example.asp

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Пример II


book store
@category="COOKING"
book
book
book
book
@lang="en"
title
auth or
year
price
title
auth or
year
price
title
auth or
year
price
title
year
price
auth or
auth or
auth or
auth or
auth or
Everyday Italian
Giada De Laurentiis
2005
30

Языки запросов к XML [Hidders and Paredaens, 2009]

- XPath (XML path language) — язык навигационных выражений
  - разрабатывался W3C, с 1999 года;
  - поиск шаблона в графе;
  - последовательность шагов для выборки узлов.
- XQuery (XML query language) — декларативный язык для запросов к коллекциям XML документов;
  - тоже W3C, 1998;
  - появился не на пустом месте: Quilt, Lorel, YATL;
  - подобен SQL, формула FLWOR;
  - включает в себя XPath.
- Множество других, нишевых: XUpdate, NEXI, XQuery Full-Text, ...

XPath

Последовательность location steps, формируем путь наподобие пути к каталогу в linux:

- Может быть абсолютным, а может быть относительным; Пример:

```
book/title/@lang
/bookstore/book/title/@lang
```

- * и @* для всех элементов и атрибутов;
- другие: //, .., |
- предикаты: 1) сравнение строк 2) существование 3) позиционные

```
//title[@lang=''en'']/..
//book[author=''J K. Rowling'' and price<30]
//book/author[2]
//book[/author]
//book[/author]/price
```

XPath: еще примеры

Запрос 1: doc("books.xml")/bookstore/book/title

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

Запрос 2: doc("books.xml")/bookstore/book[price<30]

```
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

XQuery I

FLOWR нотация:

- FOR — выборка последовательности узлов;
- LET — привязка последовательности к переменной;
- WHERE — фильтрация узлов;
- ORDER BY — сортировка узлов;
- RETURN — что возвращать (вычисляется один раз для каждого узла).

Пример 1:

```
1 for $x in doc("books.xml")/bookstore/book
2 where $x/price>30
3 return $x/title
```

Результат:

```
1 <title lang="en">XQuery Kick Start</title>
2 <title lang="en">Learning XML</title>
```

XQuery II

Пример 2:

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

Результат:

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

XPath аналог так не может:

```
doc("books.xml")/bookstore/book[price>30]/title
```

XQuery III

FLOWR нотация:

- В FOR и LET может быть несколько последовательностей;
- В RETURN может быть IF или CASE;
- В FLOWR узлы можно создавать на лету;
- ...

Пример 3:

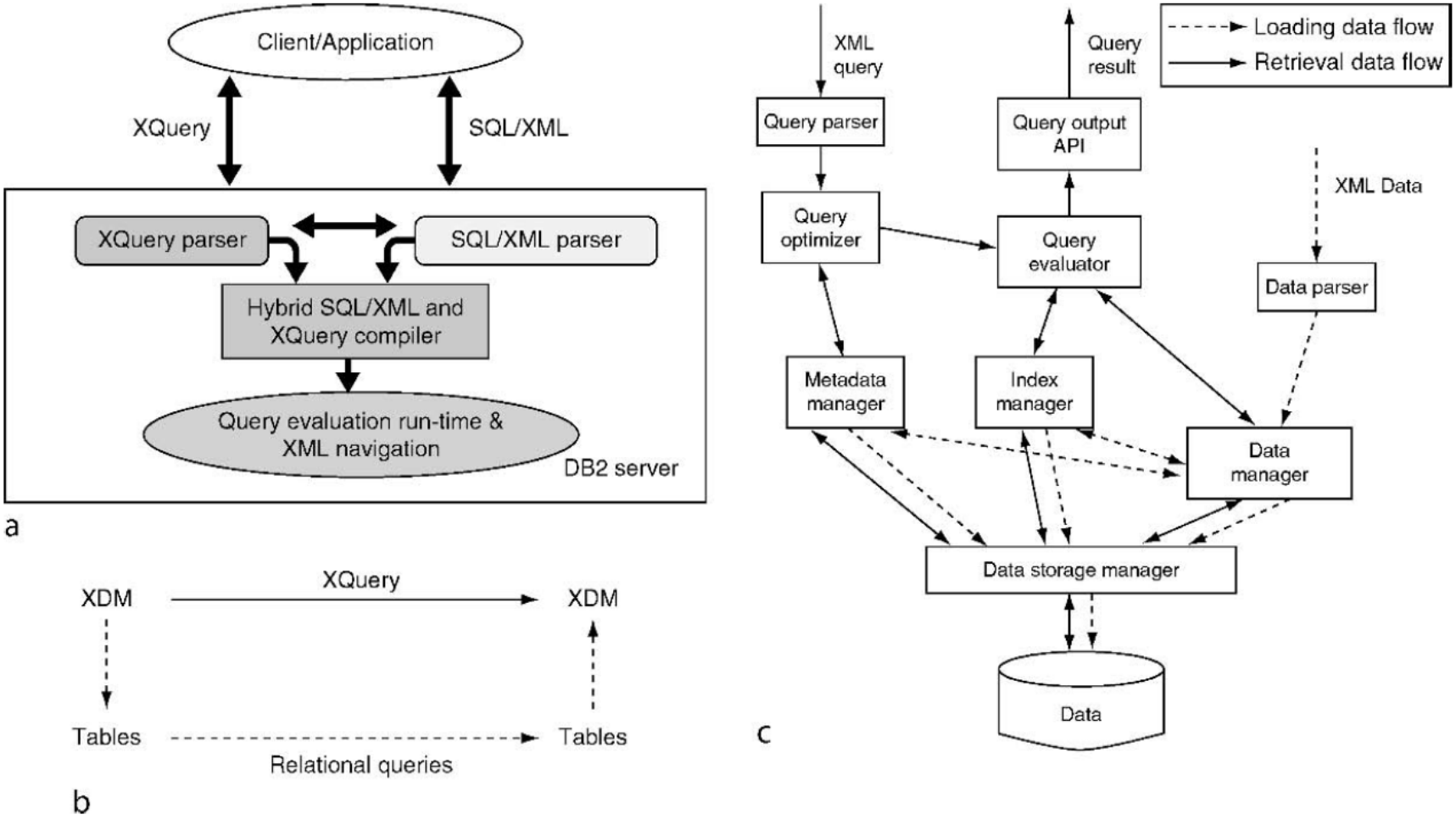
```
1 for $s in fn:doc(''students.xml'')//student,
2     $e in fn:doc(''enrollments.xml'')//enrollment
3 let $cn := fn:doc(''courses.xml'')//course[@crs-code=$e/@crs-
    code]/name
4 where $s/@stud-id = $e/@stud-id
5 order by $cn
6 return <enroll> {$s/name, $cn} </enroll>
```

Замечание: XQuery — Тьюринг-полный.

XQuery: выполнение [Grust et al., 2009]

- Использовать реляционную СУБД: оттранслировать XML в таблицы, оттранслировать запрос в SQL, выполнить, оттранслировать результаты в XML и выдать обратно;
- Выполнять нативно. Требуются специальные методы индексирования и выполнение запросов;
- Гибридная схема, смесь реляционных подходов и XML технологий.

Примеры систем



XQuery Processors. Figure 1. (a) *DB2 XML* system architecture. (b) Pathfinder: purely relational XQuery on top of vanilla database back-ends. (c) Timber architecture overview [11].

[1]

[1] Изображение взято из [Grust et al., 2009]

Реляционные СУБД и XML

Pathfinder: чисто реляционное выполнение XQuery

- Идея: существующие реляционные системы могут эффективно исполнять XQuery;
- Модель XDM (XQuery Data Model) — каждый узел это запись в таблице, возможно, с “хитрой” системой кодирований (pre/post region [2], ORDPATH) [Grust et al., 2009];
- Вычисление запроса: результаты FOR (для каждой переменной) выкладываем в таблицу, “разворачивая” их;
- Далее, на эту таблицу надстраиваем операторы специальной алгебры (Flat Relational Algebra).

[2] http://db.in.tum.de/ grust/files/xquery-on-sql-hosts.pdf материалы школы EDBT

Pathfinder: пример I

Nested `for` blocks

s_0 [`for $`v_0 `in (10,20)`
  s_1 [`for $`v_1 `in (100,200)`
    s_2 [`return $`v_0 `+ $`v_1

$loop(s_0)$

iter
1

$loop(s_1)$

iter
1
2

$loop(s_2)$

iter
1
2
3
4

- Derive $\$v_0$, $\$v_1$ as before (uses row numbering operator ϱ):

$\$v_0$ in s_1:

iter	pos	item
1	1	10
2	1	20

$\$v_1$ in s_2:

iter	pos	item
1	1	100
2	1	200
3	1	100
4	1	200

3

[3] Изображение взято из http://db.in.tum.de/~grust/files/xquery-on-sql-hosts.pdf

Pathfinder: пример II

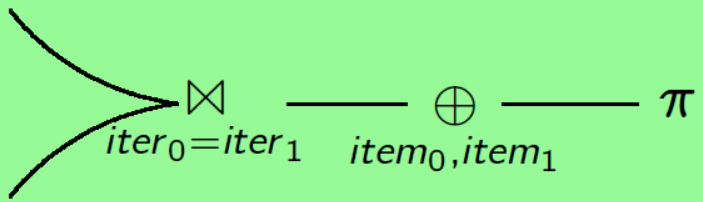
```
for $v0 in (10,20)
    for $v1 in (100,200)
      s2 [ return $v0 + $v1
```

v_0

$iter_0$	pos_0	$item_0$
1	1	10
2	1	10
3	1	20
4	1	20

v_1

$iter_1$	pos_1	$item_1$
1	1	100
2	1	200
3	1	100
4	1	200



=

iter	pos	item
1	1	110
2	1	210
3	1	120
4	1	220

4

[4] Изображение взято из http://db.in.tum.de/~grust/files/xquery-on-sql-hosts.pdf

DB2 XML: гибрид реляционной и XML СУБД

- Идея: реляционные и XML данные могут успешно дополнять друг с друга, но XML данные потребуют свого хранилища и процессора;
- Хранит двоичное представление XML дерева, не надо парсить;
- Запросы не в SQL, а в свой Query Graph Model (QGM);
- Была перезапись запроса и стоимостная оптимизация для XQuery;
- Не сильно зависит от схемы, но использует для выбора индексов;
- Индексы для XPath выражений:
  - строится для выражения, умеет /, //, [ ];
  - представлен как два B^+-дерева — индекс путей и индекс вершин;
  - индекс путей — все уникальные развернутые пути, в ID;
  - индекс вершин — ID пути в значения и ID узлов;
  - алгоритм проверки вложенности деревьев для выбора индекса;

Нативные XML СУБД [Grust et al., 2009]

- XML данные хранятся в их естественном виде, нет затрат на конвертацию в SQL и обратно;
- Примеры: Timber [Jagadish et al., 2002] и Sedna [Taranov et al., 2010]
- В Timber — своя алгебра для работы с XML: access methods, методы перезаписи запроса и стоимостная оптимизация;
- При загрузке документа узлу сопоставляется (D, S, E, L): Document, Start Key, End Key, Level (node);
- Эта схема позволяет быстро устанавливать отношения между узлами:
  - $< d_1, s_1, e_1, l_1 >$ предшествует $< d_2, s_2, e_2, l_2 > \Longleftrightarrow s_1 < s_2 \& e_1 > e_2$
- Узлы хранятся в порядке Start key.

Timber

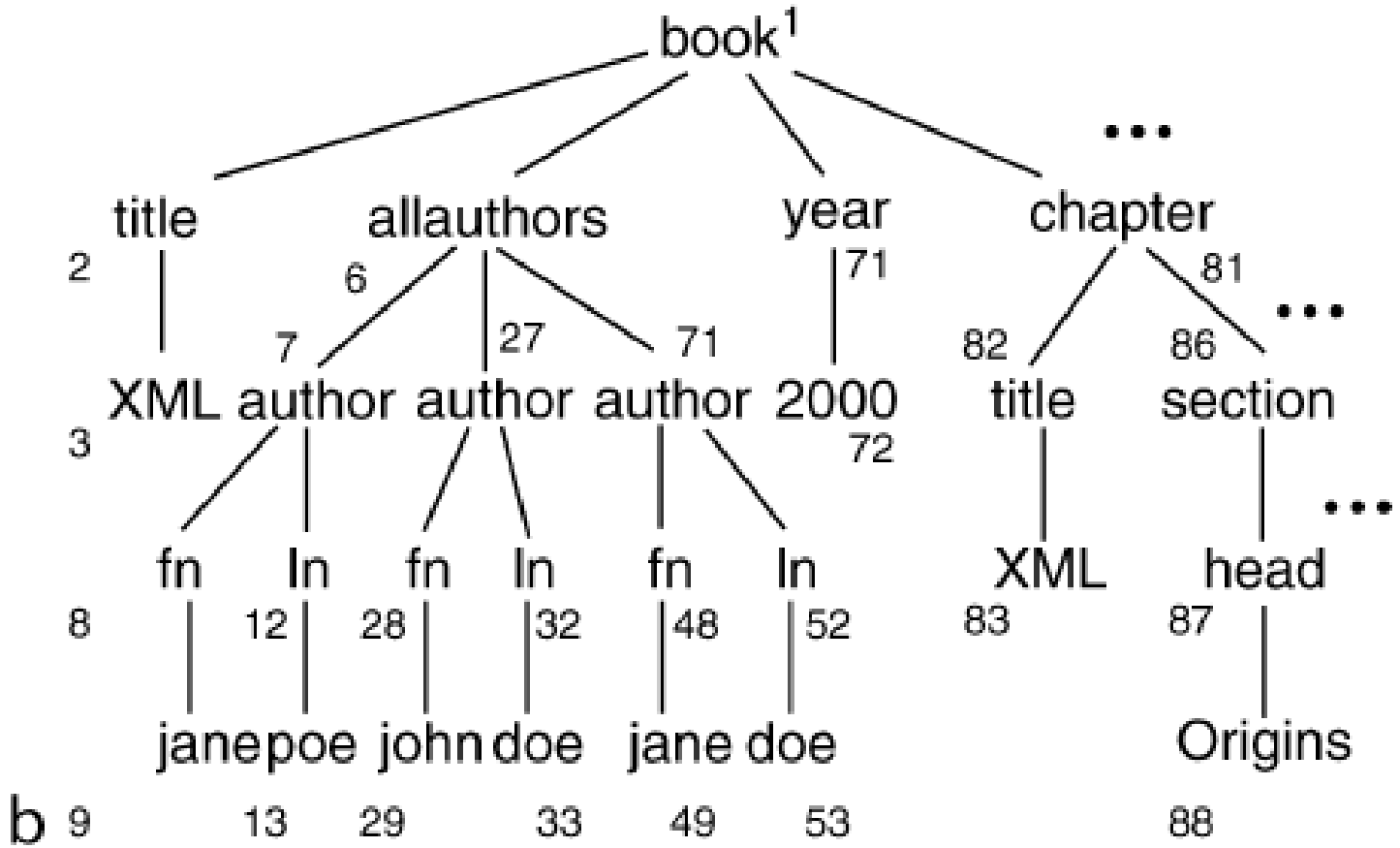
- Представление XQuery — алгебра TLC (Tree Logical Class), оперирует последовательностями упорядоченных, помеченных деревьев;
- TIX — алгебра-расширение для систем информационного поиска, добавили функции оценки;
- Новые физические операторы: материализация узла (по ID получить поддерево), структурное соединение (вычисление XPath) — набор stack-based алгоритмов;
- Оптимизатор: когда материализовывать, последовательность структурных соединений — выбор порядка;
- Оптимизатор — стоимостной, оценка результата основывается на позиционной гистограмме.

Немного про индексирование XPath

По [Luna Dong and Srivastava, 2009]:

- Схемы нумерации:
  - ID — в лоб: подобие инвертированного индекса, но это дорого :(
  - Dewey — аналог библиотечной классификации, проверка подстроки (делаем trie), плоха если дерево глубокое;
  - Интервальная схема — тройки из (первая нумерация, вторая нумерация, первая нумерация родителя) при левостороннем обходе;
    - проверять вложенность для потомка, для ребенка равенство;
    - потом на интервалах можно сделать R-tree;
- Индексирование путей: записать путь от одной вершины и до другой.

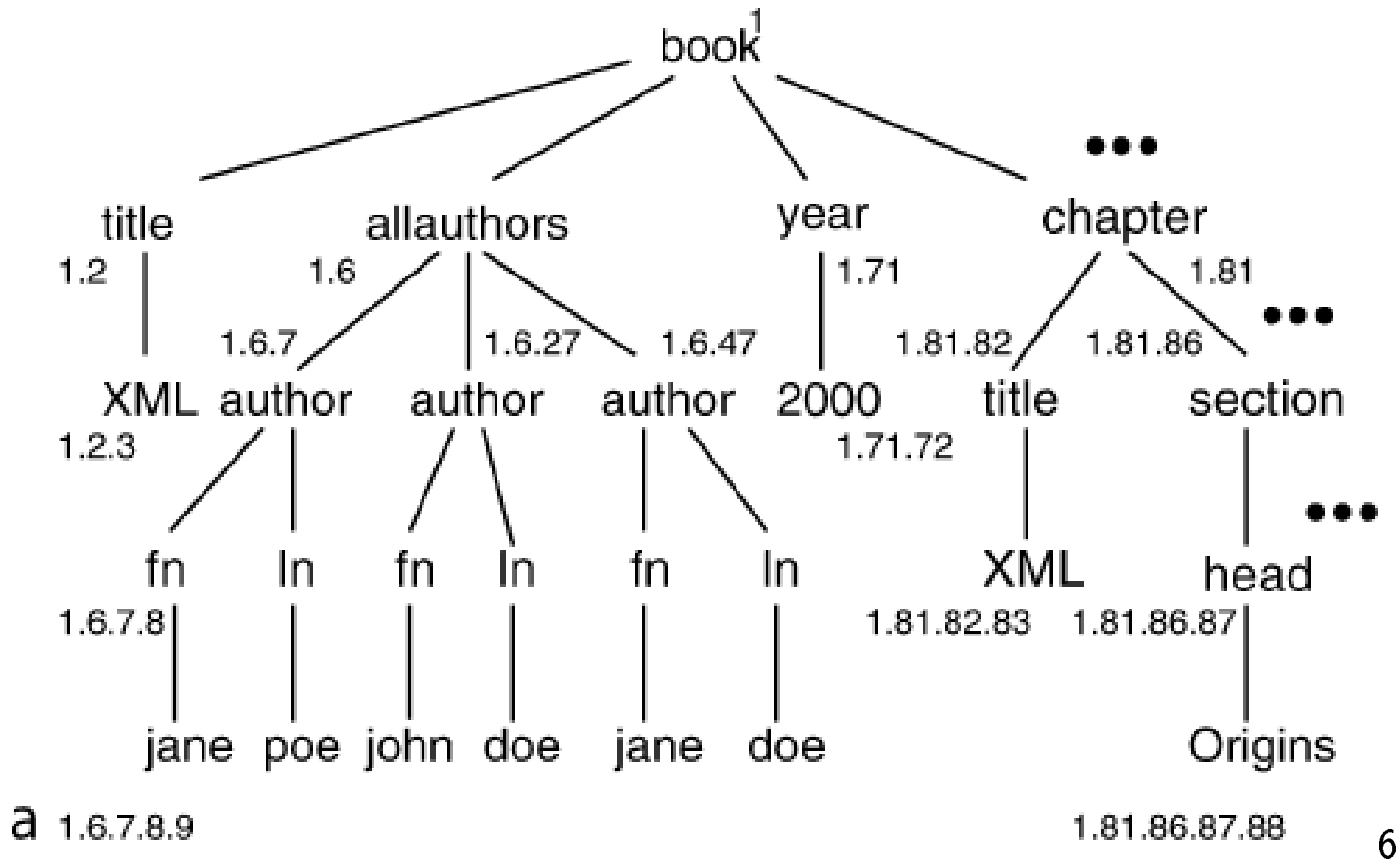
Исходные данные:



5

[5] Изображение взято из [Luna Dong and Srivastava, 2009]

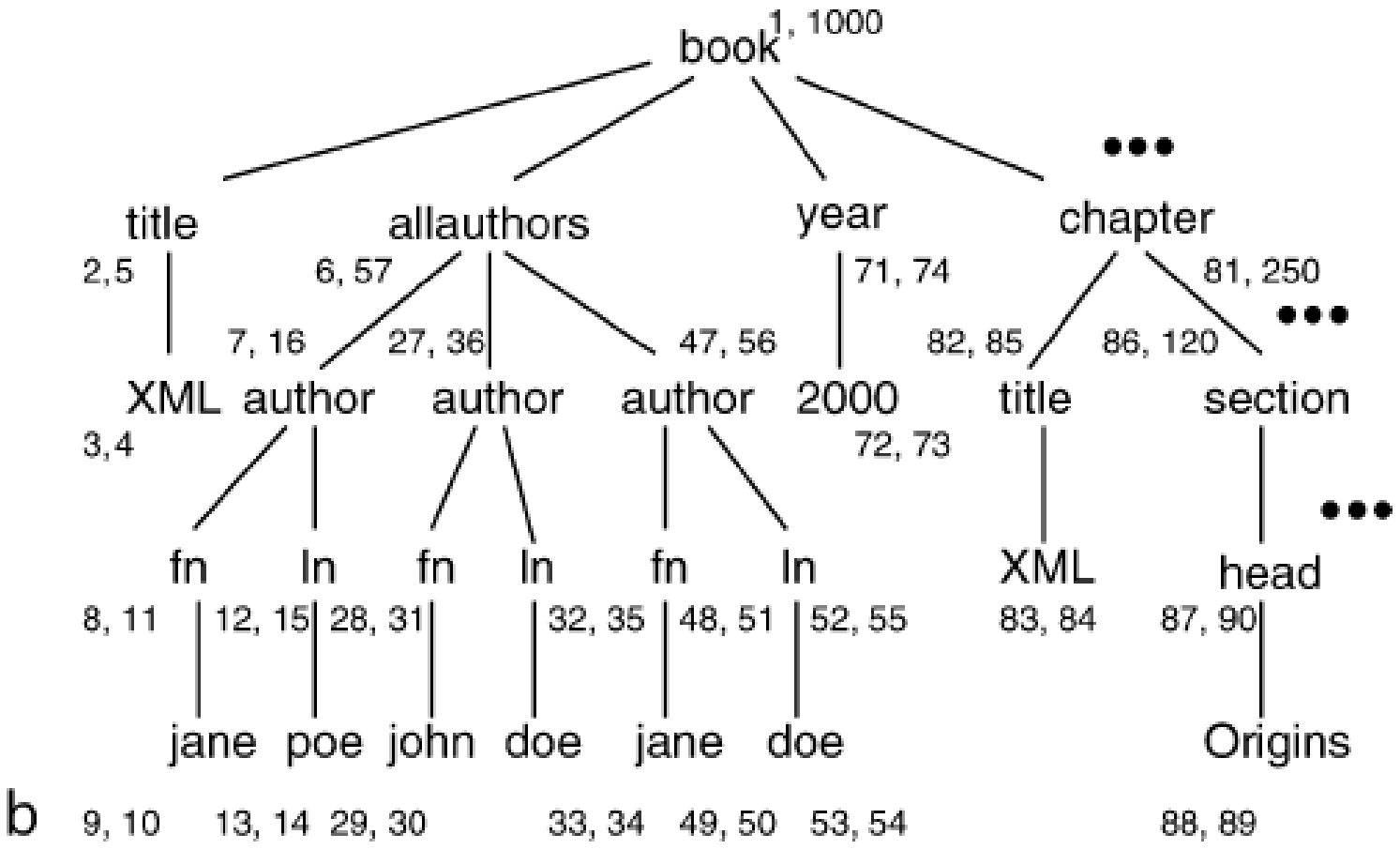
Схема Dewey



6

[6] Изображение взято из [Luna Dong and Srivastava, 2009]

Интервальная схема



7

[7] Изображение взято из [Luna Dong and Srivastava, 2009]

Схема методов для подхода индексирования путей I

XML Indexing. Table 1. The 4-ary relation for path indexes

HeadId	SchemaPath	LeafValue	IdList
1	**B**	null	[]
1	**BT**	null	[2]
1	**BT**	XML	[2]
1	**BU**	null	[6]
1	**BUA**	null	[6,7]
1	**BUAF**	null	[6,7,8]
1	**BUAF**	jane	[6,7,8]
1	**BUAL**	null	[6,7,12]
1	**BUAL**	poe	[6,7,12]
	...		
6	**U**	null	[]
6	**UA**	null	[7]
6	**UAF**	null	[7,8]
6	**UAF**	jane	[7,8]
6	**UAL**	null	[7,12]
6	**UAL**	poe	[7,12]
	...		

8

[8] Изображение взято из [Luna Dong and Srivastava, 2009]

Схема методов для подхода индексирования путей II

XML Indexing. Table 2. Members of family of path indexes

Index	Subset of `SchemaPath`	Sublist of `IdList`	Indexed Columns
Value [11]	paths of length 1	only last Id	`SchemaPath, LeafValue`
Forward link [11]	paths of length 1	only last Id	`HeadId, SchemaPath`
DataGuide [7]	root-to-leaf path prefixes	only last Id	`SchemaPath`
Index Fabric [6]	root-to-leaf paths	only first or last Id	`reverse SchemaPath, LeafValue`
ROOTPATHS [3]	root-to-leaf path prefixes	full IdList	`LeafValue, reverse SchemaPath,`
DATAPATHS [3]	all paths	full IdList	`LeafValue, HeadId, reverse SchemaPath`

9

[9] Изображение взято из [Luna Dong and Srivastava, 2009]

Ссылки I


H. V. Jagadish, S. Al-Khalifa, A. Chapman, L. V. S. Lakshmanan, A. Nierman, S. Paparizos, J. M. Patel, D. Srivastava, N. Wiwatwattana, Y. Wu, and C. Yu. 2002. TIMBER: A native XML database. The VLDB Journal 11, 4 (December 2002), 274–291. DOI=http://dx.doi.org/10.1007/s00778-002-0081-x


Xin Luna Dong, Divesh Srivastava. XML Indexing. Encyclopedia of Database Systems. Springer US, 2009. 3585–3591. http://dx.doi.org/10.1007/978-0-387-39940-9_779


Torsten Grust, H. V. Jagadish, Fatma Ozcan, Cong Yu. XQuery Processors. Encyclopedia of Database Systems. Springer US, 2009. 3671–3675.


Jan Hidders and Jan Paredaens. XPath/XQuery. Encyclopedia of Database Systems. Springer US, 2009. 3659–3665.http://dx.doi.org/10.1007/978-0-387-39940-9_774

Ссылки II

Ilya Taranov, Ivan Shcheklein, Alexander Kalinin, Leonid Novak, Sergei Kuznetsov, Roman Pastukhov, Alexander Boldakov, Denis Turdakov, Konstantin Antipin, Andrey Fomichev, Peter Pleshachkov, Pavel Velikhov, Nikolai Zavaritski, Maxim Grinev, Maria Grineva, and Dmitry Lizorkin. 2010. Sedna: native XML database management system (internals overview). In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10). ACM, New York, NY, USA, 1037-1046.
DOI=http://dx.doi.org/10.1145/1807167.1807282