

知識プログラミング方法論

第13回:XML DB

ライフスタイルデザイン研究センター 金井秀明

XML DBの基礎

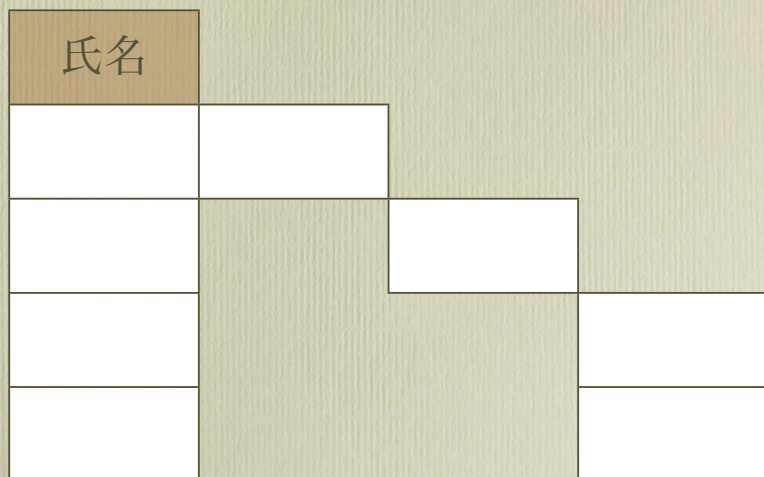
XML DBとは

- ・XMLの階層構造（半構造）をそのまま管理できる。
- ・RDBの場合、予めスキーマ（データ構造）を定義しなければならない。
- ・XML文書のノードを高速にトラバースできる。
- ・

構造化データ

name	author	publisher	price

半構造化データ



5

RDBでの対策#1

- XMLの要素データをテーブルのカラムに分解し、格納する。
- 定型的なXMLデータに対して可能な方法で、文書形式のXMLドキュメントでは適用できない。



分解の様子

```
<book>
  <Title language="JP">やさしいXML</Title>
  <Author language="JP">
    <LastName>高橋</LastName>
    <FirstName>麻奈</FirstName>
  </Author>
  <Price location="JP">2730</Price>
</book>
```

OpenXML ↓ FOR XML ↑

Title	LastName	FirstName	Price
やさしいXML	高橋	麻奈	2730

問題点：データの順序性#1

- RDBには行 (row) の間の順序はない。カラム間の前後関係もない

ページ	章	題名	内容
1	第1章	XML DBとは	...
15	第2章	RDBとXMLDB	...
50	第3章	XMLDBのクエリー言語	...
100	第4章	XMLDB関連の仕様	...

章	題名	ページ	内容
第4章	XMLDB関連の仕様	100	...
第2章	RDBとXMLDB	15	...
第1章	XML DBとは	1	...
第3章	XMLDBのクエリー言語	50	...

8

問題点：データの順序性#2

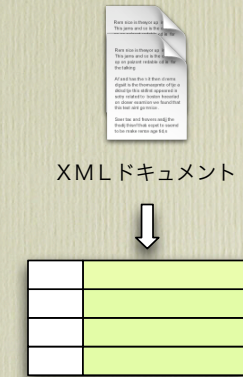
- XMLでは第1章から順に並んでいることに意味がある。

<pre><Book> <Chapter>第1章</Chapter> <Title>XML DBとは</Title> <Page>1</Page> <Chapter>第2章</Chapter> <Title>RDBとXML DBとは</Title> <Page>15</Page> <Chapter>第3章</Chapter> <Title>XMLDBのクエリー言語</Title> <Page>50</Page> <Chapter>第4章</Chapter> <Title>XMLDB関連の仕様</Title> <Page>100</Page> </Book></pre>	<pre><Book> <Chapter>第4章</Chapter> <Title>XMLDB関連の仕様</Title> <Page>100</Page> <Chapter>第2章</Chapter> <Title>RDBとXML DBとは</Title> <Page>15</Page> <Chapter>第1章</Chapter> <Title>XML DBとは</Title> <Page>1</Page> <Chapter>第3章</Chapter> <Title>XMLDBのクエリー言語</Title> <Page>50</Page> </Book></pre>
---	---

9

RDBでの対策#2

- ページサイズを大きくし、XMLデータを1つのテキストとして1つの列に保存 (CLOB: Character Large Objectの利用)
- 全文検索処理は可能だが、XML要素としての操作はできない。



RDBでの対策#3

- CLOB型をXML専用に拡張 (XMLデータ型)
 - XMLスキーマへの対応
 - XPath, XQueryのサポート
 - XML固有のインデックスのサポート
 - XQueryの拡張としての更新処理のサポート

11

ネイティブXML DB

- XML文書その構造のまま格納、操作することができるデータベース。
- 例えば、
 - Xpriori社のNeoCore (well-formedがOK)
 - <http://www.neocore.jp/xml/> <http://www.xmldb.jp/>
 - Apache Xindice (ジンディーチェ)
 - <http://xml.apache.org/xindice/>

12

Structured Query Language(SQL)

- ・ RDBにおける「データ定義」, 「データ操作」および「データ制御」のためのクエリ言語.

	author	publisher	price

13

データ定義

- ・ データベースのテーブルやオブジェクトの作成, 削除, 修正
 - ・ CREATE
 - ・ DROP
 - ・ ALTER

14

データ操作#I

- ・ 更新系
 - ・ データの挿入, 更新, 削除
- ・ 検索系
 - ・ データの検索

15

データ操作：更新系#I

- ・ 挿入

```
INSERT INTO テーブル名 (カラム名1, カラム名2, ...)  
VALUES (値1, 値2, ...)
```

```
INSERT INTO book (title, author, publisher, price)  
VALUES (やさしいXML,高橋麻奈,ソフトバンク, 2600)
```

16

データ操作：更新系#2

- 更新

```
UPDATE テーブル名 SET カラム名2=値2, カラム名3=値3  
WHERE カラム名1=値1
```

```
UPDATE book SET publisher=ソフトバンクパブリッシング,  
price=2730 WHERE title=やさしいXML
```

17

データ操作：更新系#3

- 削除

```
DELETE FROM テーブル名 WHERE カラム名1=値1
```

```
DELETE FROM book WHERE title=やさしいXML
```

18

データ操作：検索系

- 検索

```
SELECT カラム名1, カラム2, ... FROM テーブル名  
SELECT title, author FROM book
```

```
SELECT カラム名1, カラム2, ... FROM テーブル名  
WHERE カラム名1=値1
```

```
SELECT title FROM book WHERE author=高橋麻奈
```

19

XML DBでは

- データ定義言語

- ・ DTD, XML Scheme

- データ操作言語

- ・ 検索系：XPath, XQuery

- ・ 更新系：まだない。ただしDOMやSAXを利用して、更新操作を行い、再度XML-DBに格納する方法がある。

20

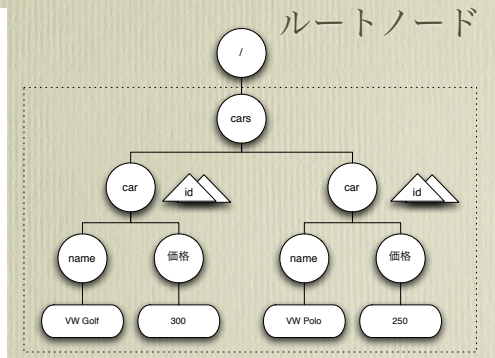
XPath#1

- XML文書の変換では、テンプレートルールによって以下のことが記述される。
 - 変換対象の指定
 - 変換内容の指定
- 変換対象の指定には、XPath: XML Path Languageが使われる。

XPath#2

- XML文書は木構造でできている。
- XPathによって、各要素を指定する。

```
<cars>
<car id="1" country="ドイツ">
  <name>VW Golf</name>
  <価格>300</価格>
</car>
<car id="2" country="ドイツ">
  <name>VW Polo</name>
  <価格>250</価格>
</car>
</cars>
```



XPath#3

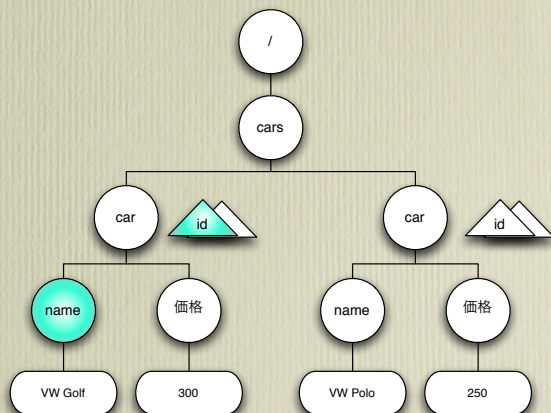
- ロケーションパス：XML文書の階層構造の中から特定のノードを指定する方法
 - 絶対：ルートノードから該当ノードまでの経路を指定する。
 - 相対：カレントノードを起点として、該当ノードまでの経路を指定する。

位置指定

- ノードの指定：「/」による階層構造
- 属性の指定：@名前
- ルートノードからの指定（絶対位置指定）
 - /: ルートノード, /cars/car/name
- .: カレントノードの指定
カレントノードがcarsのときは, car/name

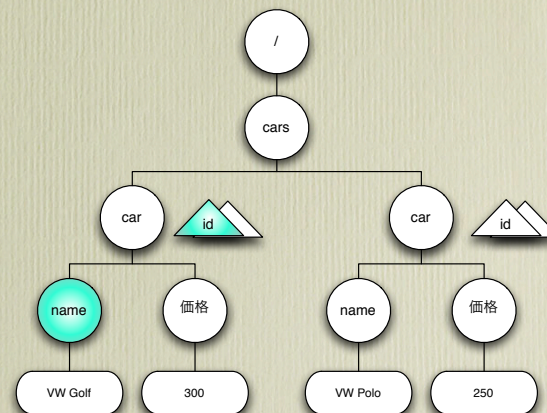
絶対位置指定の例

- /cars/car/name
- /cars/car/@id



相対位置指定の例

- 例えば、カレントノードがcarsの時
- car/name
- car/@id



データの絞り込み

- 数値で指定して、絞り込む
 - 例) 2台目の車 `car[2]`
- 演算子を使って、絞り込む
 - 条件文を記述できる.
 - 例) 価格が200以上の車
`car[price >=200]`

XQuery

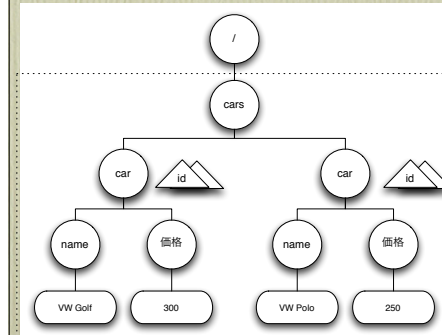
- XML DB用のクエリ言語
- XMLのツリー構造から特定の部分を抽出し、ツリー構造に整形する.
- XPathにFLWOR構文を追加したもの
 - FLOWER: 「for」「let」「where」「order by」「return」という基本構文の頭文字

FLOWR

構文	内容
FOR	反復のためのFORループを構成する
LET	変数への代入を可能にする
ORDER BY	昇順, 降順にデータをソートする
WHERE	条件を指定する
RETURN	結果の返すXMLを構成できる

29

XQueryの例#1



```
LET $cars := doc( "cars.xml")
FOR $c IN $cars/car
LET $cname := $c/name
ORDER BY $cname
RETURN $cname#
```



```
<name>VW Golf</name>
<name>VW Polo</ name>
```

30

XQueryの例#2

```
LET $book := doc( "bib.xml")
FOR $b IN $book
WHERE $b/publisher = "Adison-Wesley" AND $b/@year > 1991
ORDER BY $b/title
RETURN $b/title#
```

31

データ操作言語：更新系#1

- ・ 現在, 仕様が検討されている. 一部のデータベースではすでにドラフトに従って実装も行われている.
 - ・ XML文書のノードの挿入, 削除
 - ・ XML文書からのノードの置換, 値の置換
- ・ これらの作業でできていない.

32

データ操作言語：更新系#2

- ・DOMやSAXを利用して、XML文書の更新操作を行い、再度XML-DBに格納する方法がある。
- ・この方法だと、更新作業が頻繁に発生するような場合（定型データを扱う）には向かない。
 - ・XML文書は非定型データなので、むしろ、頻繁な更新作業は少ないと考えられる。

33

データアクセス用API

- ・RDB
 - ・Open Database Connectivity(ODBC), JDBC, .Net Framework
- ・XML DB
 - ・検討中。例えば、XQJ (XQuery API for Java)

34

プロトコル

- ・データベースとアプリケーション間通信の手順
- ・RDB
 - ・ベンダ毎の独自プロトコル
- ・XML DB
 - ・ベンダ毎の独自プロトコル
 - ・文字列でのやり取りなので、将来的には、相互接続性の高い、汎用的なものになるかも

35

RDBとXML DB

- ・RDB
 - ・構造化データが得意
 - ・参照整合性がある。
- ・XML DB
 - ・半構造化、非構造化データが得意
 - ・更新操作が多い場合には向かない。

36