How to store the XML data physically?

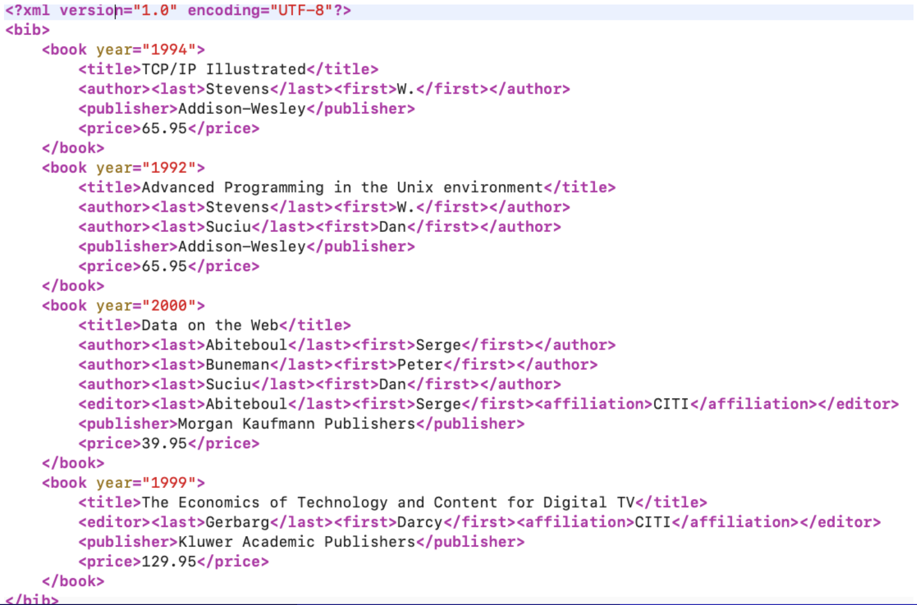
Store XML in a column

- CLOB (Character Large OBject) type + full-text indexing, or better, special XML type + functions
- Poor integration with relational query processing
- Updates are expensive

Mapping XML to relational tables

- Interval-based mapping
- Path-based mapping

XML Example



```xml
<?xml version="1.0" encoding="UTF-8"?>
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
        <author><last>Stevens</last><first>W.</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <editor><last>Abiteboul</last><first>Serge</first><affiliation>CITI</affiliation></editor>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
    </book>
    <book year="1999">
        <title>The Economics of Technology and Content for Digital TV</title>
        <editor><last>Gerbarg</last><first>Darcy</first><affiliation>CITI</affiliation></editor>
        <publisher>Kluwer Academic Publishers</publisher>
        <price>129.95</price>
    </book>
</bib>
```

XRel: A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases

The path from the root node to an element (or attribute) node can be represented by a path expression, e.g. #/bib#/book#/author

Path

PId	PathExpr
1	#/bib
2	#/bib#/book
3	#/bib#/book#/@year
4	#/bib#/book#/title
5	#/bib#/book#/author
6	#/bib#/book#/author#/last
7	#/bib#/book#/author#/first
8	#/bib#/book#/publisher
9	#/bib#/book#/price
10	#/bib#/book#/editor
11	#/bib#/book#/editor#/last
12	#/bib#/book#/editor#/first
13	#/bib#/book#/editor#/affiliation

Definition

The *region* of an element or text node is a pair of numbers that represent, respectively, the start and end positions of the node in an XML document. The region of an attribute node is a pair of two identical numbers equal to the start position of the parent element node plus 1.

The basic XRel schema consists of the following four relational schemas:

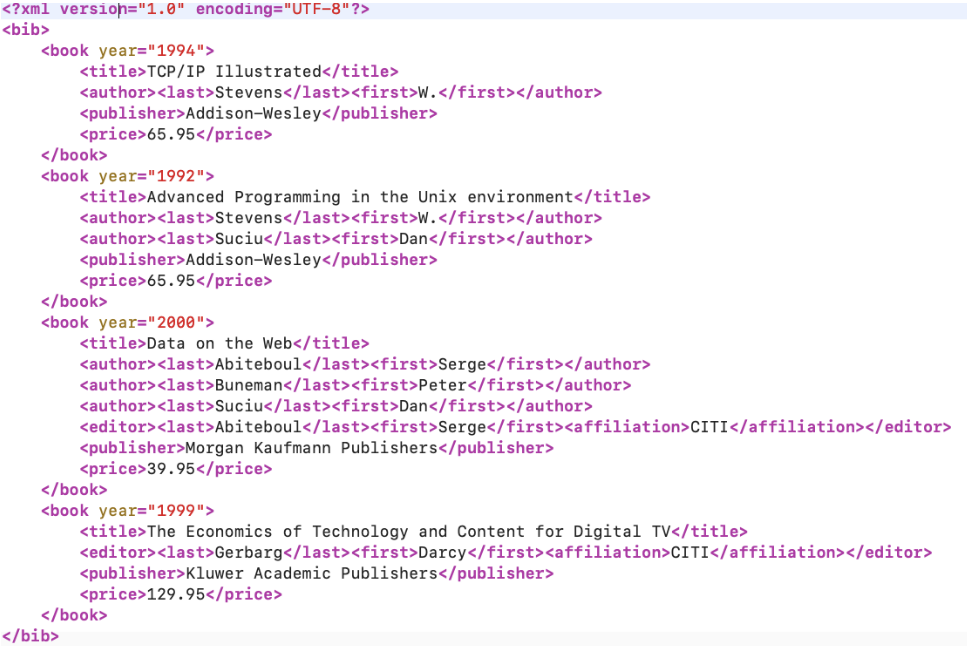
- Element(Start, End, PId)
- Attribute(Start, End, PId, Value)
- Text(Start, End, PId, Value)
- Path(PId, Pathexp).

Element

Start	End	PId
0	1058	1
5	167	2
23	48	4
56	101	5
64	76	6
84	93	7
110	135	8
...	...	...
823	884	4
892	971	10
900	913	11
920	932	12
940	957	13
980	1017	8
1029	1042	9

Attribut

PId	Start	End	Value
3	6	6	1994
16	175	175	1992
34	423	423	2000
64	806	806	1999



```
<?xml version="1.0" encoding="UTF-8"?>
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
        <author><last>Stevens</last><first>W.</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>
    <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <editor><last>Abiteboul</last><first>Serge</first><affiliation>CITI</affiliation></editor>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
    </book>
    <book year="1999">
        <title>The Economics of Technology and Content for Digital TV</title>
        <editor><last>Gerbarg</last><first>Darcy</first><affiliation>CITI</affiliation></editor>
        <publisher>Kluwer Academic Publishers</publisher>
        <price>129.95</price>
    </book>
</bib>
```

Text

Start	End	Value	PId
30	47	TCP/IP Illustrated	4
70	76	Stevens	6
91	92	W.	7
121	134	Addison-Wesley	8
154	158	65.95	9
199	242	Advanced Programming in the Unix environment	4
265	271	Stevens	6
286	287	W.	7
...	...	...	...
740	765	Morgan Kaufmann Publishers	8
785	789	39.95	62
830	883	The Economics of Technology and Content for Digital TV	4
906	912	Gerbarg	11
927	931	Darcy	12
953	956	CITI	13
991	1016	Kluwer Academic Publishers	8
1036	1041	129.95	9

```
//author/last

SELECT  Text.Value
FROM Element E, Text T, Path P
WHERE P.PathExpr like ’#%/author#/last’
   AND E.pId = P.PId
   AND T.Start > E.Start
   AND T.End < E.End
```

Output all the books published by "Addison-Wesley".

```
//book[./publisher = "Addison-Wesley"]
```

```
SELECT E1.Start, E1.End Book
FROM Element E1, Element E2, Text T, Path P1, Path P2
WHERE P1.PathExpr like '#%/book'
   AND E1.pId = P1.PId AND E1.START < E2.START AND E1.END > E2.END
   AND P2.PathExpr like '#%/book#/publisher'
   AND E2.pId = P2.PId
   AND T.Value = "Addison-Wesley"
   AND T.Start > E2.Start
   AND T.End < E2.End
```

Output all the books' titles that are published by "Addison-Wesley".

Summary

- XML data can be “shredded” into rows in a relational database
- Queries can then benefit from smart relational indexing, optimization, and execution
- Different data mapping techniques lead to different styles of queries