Lecture 11: Relational Algebra, XML and Semistructured Data

Wednesday, October 23, 2002

XML: Outline

XML

- Relational algebra (5.2, 5.3, 5.4)
- XML (4.6, 4.7)
  - This lecture: syntax, semistructured data
  - Next lectures: DTDs, XPath, XQuery
- Additional readings for XML:
  http://www.w3.org/XML/1999/XML-in-10-points
  www.zvon.org/xxl/XMLTutorial/General/book_en.html
- Main source: www.w3.org (but hard to read)

Relational Algebra

- Five operators:
  - Union: $\cup$
  - Difference: -
  - Selection: σ
  - Projection: Π
  - Cartesian Product: $\times$
- Derived or auxiliary operators:
  - Intersection, complement
  - Joins (natural,equi-join, theta join, semi-join)
  - Renaming: ρ

Renaming

- Changes the schema, not the instance
- Notation: $\rho_{B1,\ldots,Bn}(R)$
- Example:
  - $\rho_{LastName,\ SocSocNo}$ (Employee)
  - Output schema:
    Answer(LastName, SocSocNo)

Renaming Example

Employee

Name	SSN
John	999999999
Tony	777777777

$\rho_{LastName,\ SocSocNo}$ (**Employee**)

LastName	SocSocNo
John	999999999
Tony	777777777

Natural Join

- Notation: R1 $\bowtie$ R2
- Meaning: R1 $\bowtie$ R2 = $\Pi_A(\sigma_C(R1 \times R2))$
- Where:
  - The selection σ_C checks equality of all common attributes
  - The projection eliminates the duplicate common attributes

Natural Join Example

Employee

Name	SSN
John	999999999
Tony	777777777

Dependents

SSN	Dname
999999999	Emily
777777777	Joe

Employee $\bowtie$ **Dependents =**

$\Pi_{Name, SSN, Dname}(\sigma_{SSN=SSN2}(Employee \times \rho_{SSN2, Dname}(Dependents))$

Name	SSN	Dname
John	999999999	Emily
Tony	777777777	Joe

7

Natural Join

- R=

A	B
X	Y
X	Z
Y	Z
Z	V

S=

B	C
Z	U
V	W
Z	V

- R ⋈ S=

A	B	C
X	Z	U
X	Z	V
Y	Z	U
Y	Z	V
Z	V	W

8

Natural Join

- Given the schemas R(A, B, C, D), S(A, C, E), what is the schema of R ⋈ S ?
- Given R(A, B, C), S(D, E), what is R ⋈ S ?
- Given R(A, B), S(A, B), what is R ⋈ S ?

9

Theta Join

- A join that involves a predicate
- $R1 \bowtie_\theta R2 = \sigma_\theta (R1 \times R2)$
- Here θ can be any condition

10

Eq-join

- A theta join where θ is an equality
- $R1 \bowtie_{A=B} R2 = \sigma_{A=B} (R1 \times R2)$
- Example:
  - Employee $\bowtie_{SSN=SSN}$ Dependents
- Most useful join in practice

11

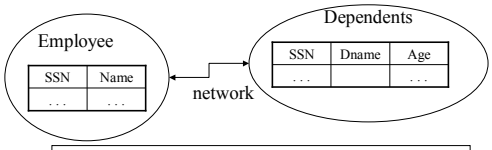
Semijoin

- $R \ltimes S = \Pi_{A1,\ldots,An} (R \bowtie S)$
- Where $A_1, \ldots, A_n$ are the attributes in R
- Example:
  - Employee ⋉ Dependents

12

Semijoins in Distributed Databases

- Semijoins are used in distributed databases



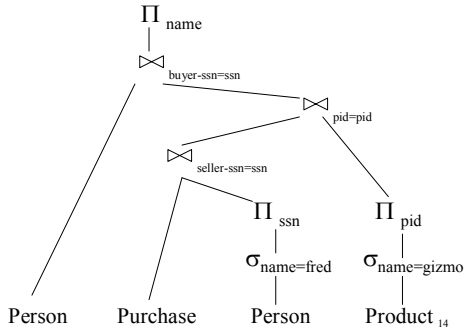
Employee $\bowtie_{ssn=ssn}$ ($\sigma_{age>71}$ (Dependents))

R = Employee $\ltimes$ T

T = Π_{SSN} $\sigma_{age>71}$ (Dependents)

Answer = R $\bowtie$ Dependents

Complex RA Expressions



Operations on Bags

A **bag** = a set with repeated elements

All operations need to be defined carefully on bags

- {a,b,b,c}$\cup${a,b,b,b,e,f,f}={a,a,b,b,b,b,b,c,e,f,f}
- {a,b,b,b,c,c} – {b,c,c,c,d} = {a,b,b,d}
- σ_C(R): preserve the number of occurrences
- Π_A(R): no duplicate elimination
- Cartesian product, join: no duplicate elimination

Important ! Relational Engines work on bags, not sets !

Reading assignment: 5.3 – 5.4

Finally: RA has Limitations !

- Cannot compute "transitive closure"

Name1	Name2	Relationship
Fred	Mary	Father
Mary	Joe	Cousin
Mary	Bill	Spouse
Nancy	Lou	Sister

- Find all direct and indirect relatives of Fred
- Cannot express in RA !!! Need to write C program

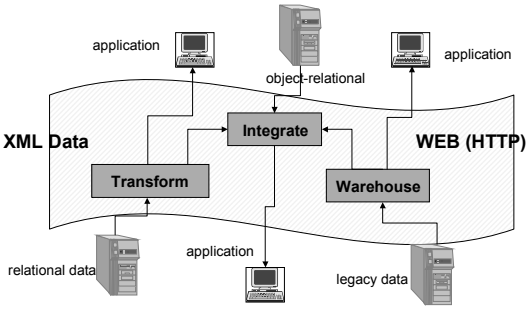
XML

- eXtensible Markup Language
- XML 1.0 – a recommendation from W3C, 1998
- Roots: SGML (a very nasty language).
- After the roots: a format for sharing *data*

Why XML is of Interest to Us

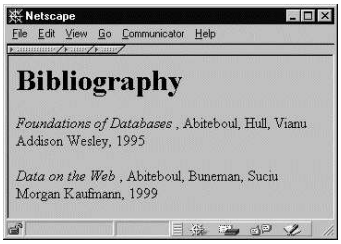
- XML is just syntax for data
  - Note: we have no syntax for relational data
  - But XML is not relational: *semistructured*
- This is exciting because:
  - Can translate *any* data to XML
  - Can ship XML over the Web (HTTP)
  - Can input XML into any application
  - Thus: data sharing and exchange on the Web

XML Data Sharing and Exchange



Specific data management tasks

From HTML to XML



HTML describes the presentation

HTML

```
<h1> Bibliography </h1>
<p> <i> Foundations of Databases </i>
      Abiteboul, Hull, Vianu
      <br> Addison Wesley, 1995
<p> <i> Data on the Web </i>
      Abiteoul, Buneman, Suciu
      <br> Morgan Kaufmann, 1999
```


XML

```
<bibliography>
    <book>   <title> Foundations… </title>
             <author> Abiteboul </author>
             <author> Hull </author>
             <author> Vianu </author>
             <publisher> Addison Wesley </publisher>
             <year> 1995 </year>
    </book>
    …
</bibliography>
```

XML describes the content

XML Terminology

- tags: book, title, author, …
- start tag: <book>, end tag: </book>
- elements: <book>…<book>,<author>…</author>
- elements are nested
- empty element: <red></red> abbrv. <red/>
- an XML document: single *root element*

well formed XML document: if it has matching tags

More XML: Attributes

```
<book price = "55" currency = "USD">
  <title> Foundations of Databases </title>
  <author> Abiteboul </author>
   …
  <year> 1995 </year>
</book>
```

attributes are alternative ways to represent data

More XML: Oids and References

```
<person id="o555">  <name> Jane </name> </person>

<person id="o456">  <name> Mary </name>
                    <children idref="o123 o555"/>
</person>

<person id="o123" mother="o456"><name>John</name>
</person>
```

oids and references in XML are just syntax

25

More XML: CDATA Section

- Syntax: <![CDATA[.....any text here...]]>
- Example:

```
<example>
   <![CDATA[ some text here </notAtag> <>]]>
</example>
```

26

More XML: Entity References

- Syntax: &entityname;
- Example:
  <element> this is less than < </element>
- Some entities:

<	<
>	>
&	&
'	'
"	"
&	Unicode char

27

More XML: Processing Instructions

- Syntax: <?target argument?>
- Example:

```
•<product> <name> Alarm Clock </name>
            <?ringBell 20?>
            <price> 19.99 </price>
</product>
```

- What do they mean ?

28

More XML: Comments

- Syntax <!-- .... Comment text... -->
- Yes, they are part of the data model !!!

29

XML Namespaces

- http://www.w3.org/TR/REC-xml-names (1/99)
- name ::= [prefix:]localpart

```
<book xmlns:isbn="www.isbn-org.org/def">
    <title> … </title>
    <number> 15 </number>
    <isbn:number> …. </isbn:number>
</book>
```

30

XML Namespaces

- syntactic: <number> , <isbn:number>
- semantic: provide URL for schema

```
<tag  xmlns:mystyle = "http://…">
        …
        <mystyle:title> … </mystyle:title>
        <mystyle:number> …
  </tag>
```

Belong to this namespace

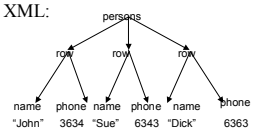
31

From Relational Data to XML Data

persons

name	phone
John	3634
Sue	6343
Dick	6363

XML:




```
<persons>
    <row> <name>John</name>
          <phone> 3634</phone></row>
    <row> <name>Sue</name>
          <phone> 6343</phone>
    <row> <name>Dick</name>
          <phone> 6363</phone></row>
</persons>
```

XML Data

- XML is self-describing
- Schema elements become part of the data
  - Reational schema: persons(name,phone)
  - In XML <persons>, <name>, <phone> are part of the data, and are repeated many times
- Consequence: XML is much more flexible
- XML = semistructured data

33

Semi-structured Data Explained

- Missing attributes:

```
<person>  <name> John</name>
          <phone>1234</phone>
 </person>

<person>  <name>Joe</name>
</person>
```

← no phone !

- Could represent in a table with nulls

name	phone
John	1234
Joe	-

34

Semi-structured Data Explained

- Repeated attributes

```
<person> <name> Mary</name>
         <phone>2345</phone>
         <phone>3456</phone>
</person>
```

← two phones !

- Impossible in tables:

name	phone	
Mary	2345	3456

???

35

Semistructured Data Explained

- Attributes with different types in different objects

```
<person> <name>  <first> John </first>
                 <last> Smith </last>
         </name>
         <phone>1234</phone>
</person>
```

← structured name !

- Nested collections (no 1NF)
- Heterogeneous collections:
  - <db> contains both <book>s and <publisher>s

36