ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

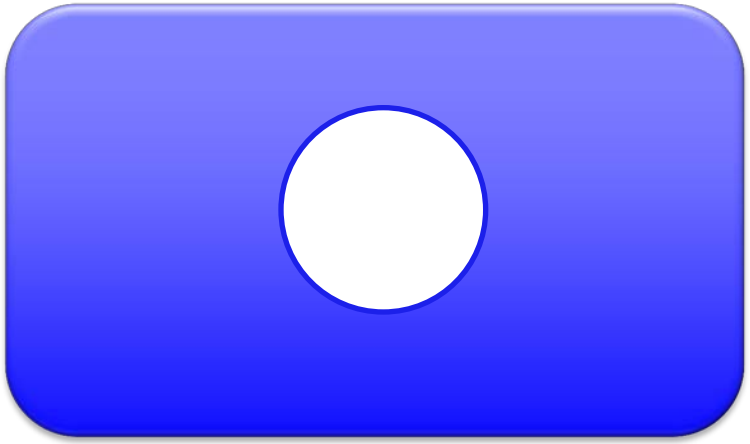
The Systems Group
at ETH Zurich

XML and Databases
Exercise Session 12

courtesy of Ghislain Fourny

Where we are: XQuery Implementation

The problem we are solving today



Square peg

Round hole

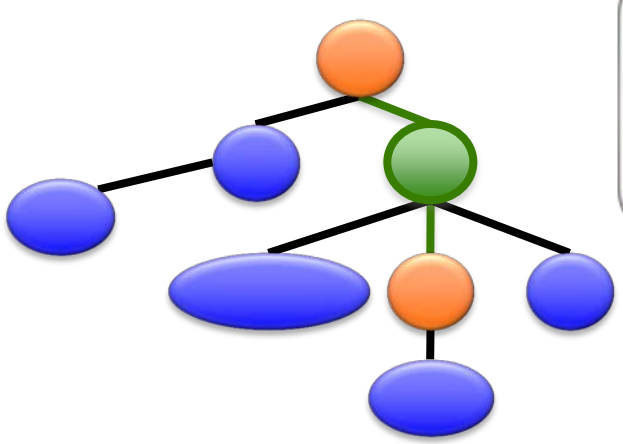
The problem we are solving today

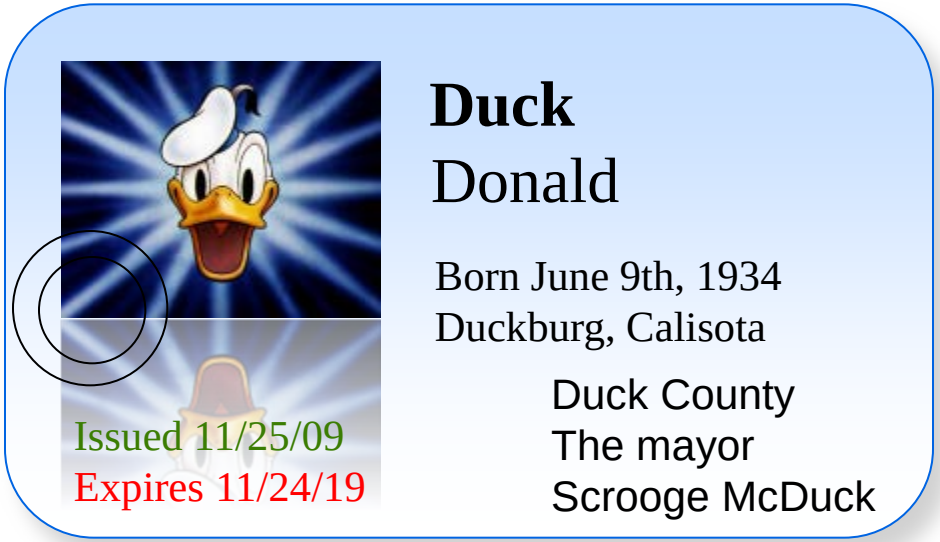
The problem we are solving today: Storing XML in an RDBMS

```
<root>Some <b>XML</b> document</root>
```

Relational Database

Good news: we will use node IDs (remember last week)





XML Documents: no nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<doc
  xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    flights_no_nesting.xsd">
"
  <!-- Airports -->
  <Airport airId="LHR">
    <name>London Heathrow</name>
    <tax>100</tax>
  </Airport>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
  <Airport airId="PAR">
    <name>Paris</name>
    <tax>100</tax>
  </Airport>

  <!-- Flights -->
  <Flight flightId="LX183">
    <seats>100</seats>
    <date>2006-12-24</date>
    <departure>08:15:00</departure>
    <arrival>10:01:00</arrival>
    <source>ZRH</source>
    <destination>PAR</destination>
  </Flight>

  <Flight flightId="LX124">
    <seats>100</seats>
    <date>2006-12-24</date>
    <departure>12:00:00</departure>
    <arrival>13:00:00</arrival>
    <source>PAR</source>
    <destination>LHR</destination>
  </Flight>

  <!-- Passengers  (there are more Santas :)
  -->
  <Passenger>
    <name>Santa Claus</name>
    <passportnumber>000111</passportnumber>
    <address>Somewhere</address>
  </Passenger>

  <Passenger>
    <name>Santa Claus</name>
    <passportnumber>000112</passportnumber>
    <address>Somewhere</address>
  </Passenger>

  <Reservation>
    <flightRef>LX124</flightRef>
    <passRef>000111</passRef>
  </Reservation>

  <Reservation>
    <flightRef>LX183</flightRef>
    <passRef>000112</passRef>
  </Reservation>
</doc>
```

XML Documents: with nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<doc

xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:noNamespaceSchemaLocation="file:/C:/Doc
uments/Teaching/WS06_07/XML06_07/ExSheet7/F
lightsSchema_nested.xsd">

    <!-- Airports -->
    <Airport airId="LHR">
        <name>London Heathrow</name>
        <tax>100</tax>
    </Airport>
    <Airport airId="ZRH">
        <name>Zurich</name>
        <tax>150</tax>
    </Airport>
    <Airport airId="PAR">
        <name>Paris</name>
        <tax>100</tax>
    </Airport>

    <!-- Flights -->
    <Flight flightId="LX183">
        <seats>100</seats>
        <date>2006-12-24</date>
        <departure>08:15:00</departure>
        <arrival>10:01:00</arrival>
        <!-- source -->
        <Airport airId="ZRH">
          <name>Zurich</name>
          <tax>150</tax>
        </Airport>
        <!-- destination -->
        <Airport airId="PAR">
          <name>Paris</name>
          <tax>100</tax>
        </Airport>
     </Flight>

    <Flight flightId="LX124">
        <seats>100</seats>
        <date>2006-12-24</date>
        <departure>12:00:00</departure>
        <arrival>13:00:00</arrival>
        <!-- source -->
        <Airport airId="PAR">
          <name>Paris</name>
          <tax>100</tax>
        </Airport>
        <!-- destination -->
        <Airport airId="LHR">
          <name>London Heathrow</name>
          <tax>100</tax>
        </Airport>
    </Flight>


    <!-- Passengers  (there are more Santas
:)       -->
    <Passenger>
        <name>Santa Claus</name>

<passportnumber>000111</passportnumber>
        <address>Somewhere</address>
    </Passenger>

    <Passenger>
        <name>Santa Claus</name>

<passportnumber>000112</passportnumber>
        <address>Somewhere</address>
    </Passenger>

    <Reservation>
        <date>2006-12-24</date>
          <Flight flightId="LX124">
            <seats>100</seats>
            <departure>12:00:00</departure>
            <arrival>13:00:00</arrival>
            <!-- source -->
              <Airport airId="PAR">
                <name>Paris</name>
                <tax>100</tax>
              </Airport>
            <!-- destination -->
              <Airport airId="LHR">
                <name>London
Heathrow</name>
                <tax>100</tax>
              </Airport>
          </Flight>
          <Passenger>
            <name>Santa Claus</name>

<passportnumber>000111</passportnumber>
            <address>Somewhere</address>
          </Passenger>
      </Reservation>

      <Reservation>
        <date>2006-12-24</date>
          <Flight flightId="LX183">
            <seats>100</seats>
            <departure>08:15:00</departure>
            <arrival>10:01:00</arrival>
            <!-- source -->
            <Airport airId="ZRH">
              <name>Zurich</name>
              <tax>150</tax>
            </Airport>
            <!-- destination -->
            <Airport airId="PAR">
              <name>Paris</name>
              <tax>100</tax>
            </Airport>
          </Flight>
          <Passenger>
            <name>Santa Claus</name>

<passportnumber>000112</passportnumber>
            <address>Somewhere</address>
          </Passenger>
    </Reservation>
</doc>
```

How to put XML in a relational DB?

- With a schema:
  - Schema-based shredding
- Without schema (generic):
  - Tree encoding
  - Edge approach

Schema-based Shredding without nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<doc
  xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    flights_no_nesting.xsd">
"
  <!-- Airports -->
  <Airport airId="LHR">
    <name>London Heathrow</name>
    <tax>100</tax>
  </Airport>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
  <Airport airId="PAR">
    <name>Paris</name>
    <tax>100</tax>
  </Airport>

  <!-- Flights -->
  <Flight flightId="LX183">
    <seats>100</seats>
    <date>2006-12-24</date>
    <departure>08:15:00</departure>
    <arrival>10:01:00</arrival>
    <source>ZRH</source>
    <destination>PAR</destination>
  </Flight>

  <Flight flightId="LX124">
    <seats>100</seats>
    <date>2006-12-24</date>
    <departure>12:00:00</departure>
    <arrival>13:00:00</arrival>
    <source>PAR</source>
    <destination>LHR</destination>
  </Flight>

  <!-- Passengers  (there are more Santas :)
  -->
  <Passenger>
    <name>Santa Claus</name>
    <passportnumber>000111</passportnumber>
    <address>Somewhere</address>
  </Passenger>

  <Passenger>
    <name>Santa Claus</name>
    <passportnumber>000112</passportnumber>
    <address>Somewhere</address>
  </Passenger>

  <Reservation>
    <date>2006-12-24</date>
    <flightRef>LX124</flightRef>
    <passRef>000111</passRef>
  </Reservation>

  <Reservation>
    <date>2006-12-24</date>
    <flightRef>LX183</flightRef>
    <passRef>000112</passRef>
  </Reservation>
</doc>
```

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Schema-based Shredding without nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<doc
  xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    flights_no_nesting.xsd">
"
  <!-- Airports -->
  <Airport airId="LHR">
    <name>London Heathrow</name>
    <tax>100</tax>
  </Airport>
  <Airport airId="ZRH">
    <name>Z
...
    ...06:15:00</departure>
    <arrival>10:01:00</arrival>
    <source>ZRH</source>
    <destination>PAR</destination>
  </Flight>

  <Flight flightId="LX124">
    <seats>100</seats>
    <date>2006-12-24</date>
    <departure>12:00:00</departure>
    <arrival>13:00:00</arrival>
    <source>PAR</source>
    <destination>LHR</destination>
  </Flight>
```

```
  <!-- Passengers  (there are more Santas :)
-->
  <Passenger>
    <name>Santa Claus</name>
    <passportnumber>000111</passportnumber>
    <address>Somewhere</address>
  </Passen...
...
    ...name>
    ...112</passportnumber>
    .../address>
...
    ...e>
    ...ghtRef>
    ...Ref>
...
  <Reservation>
    <date>2006-12-24</date>
    <flightRef>LX183</flightRef>
    <passRef>000112</passRef>
  </Reservation>
</doc>
```

These are (almost) already database tables!

Schema-based Shredding without nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<doc
  xmlns:xsi=
    "http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    flights_no_nesting.xsd">
"
<!-- Flights -->
<Flight flightId="LX183">
  <seats>100</seats>
  <date>2006-12-24</date>
  <departure>08:15:00</departure>
  <arrival>10:01:00</arrival>
  <source>ZRH</source>
  <destination>PAR</destination>
</Flight>
```

Doc

airId	name	tax
LHR	London Heathrow	100
ZRH	Zurich	150
PAR	Paris	100

```
  <!-- Passengers  (there are more Santas :)
-->
  <Reservation>
    <fli
    <pas
  </Rese
  <Reser
    <fli
    <pas
  </Rese
</doc>
```

Passenger

name	passportnumber	address
Santa Claus	000111	Somewhere
Santa Claus	000112	Somewhere

Reservation

flightRef	passRef
LX124	000111
LX183	000112

Flight

flightId	seats	date	departure	arrival	source	destination
LX183	100	2006-12-24	08:15:00	10:01:00	ZRH	PAR
LX124	100	2006-12-24	12:00:00	13:00:00	PAR	ZRH

Schema-based Shredding without nesting

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSche
ma">

  <xs:element name="Passenger">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name"
type="xs:string"/>
        <xs:element name="passportnumber"
type="xs:string"/>
        <xs:element name="address"
type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Flight">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="seats">
          ...
        </xs:element>
        <xs:element name="date"
type="xs:date"/>
        <xs:element name="departure"
type="xs:time"/>
        <xs:element name="arrival"
type="xs:time"/>
        <xs:element name="source"
type="xs:string"/>
        <xs:element name="destination"
type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="flightId">
        ...
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="Airport">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name"
type="xs:string"/>
        <xs:element name="tax"
type="xs:float"/>
      </xs:sequence>
      <xs:attribute name="airId">
        ...
      </xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="Reservation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="flightRef"
type="xs:string"/>
        <xs:element name="passRef"
type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="doc">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Airport"
minOccurs="2" maxOccurs="unbounded" />
        <xs:element ref="Flight"
minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="Passenger"
minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="Reservation"
minOccurs="1" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>

  </xs:element>
</xs:schema>
```

Schema-based Shredding without nesting: the DB schema

Passenger(passportNumber, name, address)
Airport(airpId, name, tax)
Flight(flightId, date, departure, arrival, seats, source, destination)
Reservation(flightRef, passRef)

Schema-based Shredding with nesting (general case)

```
<?xml version="1.0" encoding="UTF-8"?>
<doc

xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"

xsi:noNamespaceSchemaLocation="file:/C:/Doc
uments/Teaching/WS06_07/XML06_07/ExSheet7/F
lightsSchema_nested.xsd">

    <!-- Airports -->
    <Airport airId="LHR">
        <name>London Heathrow</name>
        <tax>100</tax>
    </Airport>
    <Airport airId="ZRH">
        <name>Zurich</name>
        <tax>150</tax>
    </Airport>
    <Airport airId="PAR">
        <name>Paris</name>
        <tax>100</tax>
    </Airport>

    <!-- Flights -->
    <Flight flightId="LX183">
        <seats>100</seats>
        <date>2006-12-24</date>
        <departure>08:15:00</departure>
        <arrival>10:01:00</arrival>
        <!-- source -->
        <Airport airId="ZRH">
          <name>Zurich</name>
          <tax>150</tax>
        </Airport>
        <!-- destination -->
        <Airport airId="PAR">
          <name>Paris</name>
          <tax>100</tax>
        </Airport>
     </Flight>
```

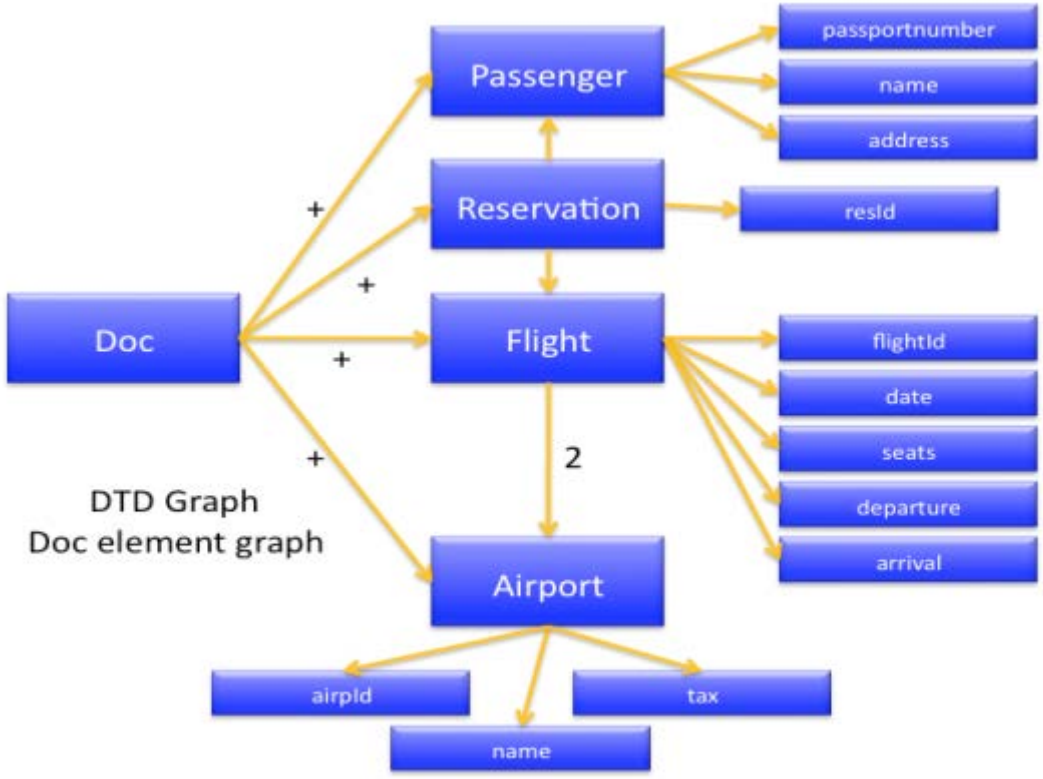
```
    <Flight flightId="LX124">
        <seats>100</seats>
        <date>2006-12-24</date>
        <departure>12:00:00</departure>
        <arrival>13:00:00</arrival>
        <!-- source -->
        <Airport airId="PAR">
          <name>Paris</name>
          <tax>100</tax>
        </Airport>
        <!-- destination -->
        <Airport airId="LHR">
          <name>London Heathrow</name>
          <tax>100</tax>
        </Airport>
    </Flight>


    <!-- Passengers  (there are more Santas
:)        -->
    <Passenger>
        <name>Santa Claus</name>

<passportnumber>000111</passportnumber>
        <address>Somewhere</address>
    </Passenger>

    <Passenger>
        <name>Santa Claus</name>

<passportnumber>000112</passportnumber>
        <address>Somewhere</address>
    </Passenger>
```

```
    <Reservation>
        <date>2006-12-24</date>
          <Flight flightId="LX124">
            <seats>100</seats>
            <date>2006-12-24</date>
            <departure>12:00:00</departure>
            <arrival>13:00:00</arrival>
            <!-- source -->
              <Airport airId="PAR">
                <name>Paris</name>
                <tax>100</tax>
              </Airport>
            <!-- destination -->
              <Airport airId="LHR">
                <name>London
Heathrow</name>
                <tax>100</tax>
              </Airport>
          </Flight>
          <Passenger>
            <name>Santa Claus</name>

<passportnumber>000111</passportnumber>
            <address>Somewhere</address>
          </Passenger>
      </Reservation>

      <Reservation>
        <date>2006-12-24</date>
          <Flight flightId="LX183">
            <seats>100</seats>
            <date>2006-12-24</date>
            <departure>08:15:00</departure>
            <arrival>10:01:00</arrival>
            <!-- source -->
            <Airport airId="ZRH">
              <name>Zurich</name>
              <tax>150</tax>
            </Airport>
            <!-- destination -->
            <Airport airId="PAR">
              <name>Paris</name>
              <tax>100</tax>
            </Airport>
          </Flight>
          <Passenger>
            <name>Santa Claus</name>

<passportnumber>000112</passportnumber>
            <address>Somewhere</address>
          </Passenger>
    </Reservation>
</doc>
```

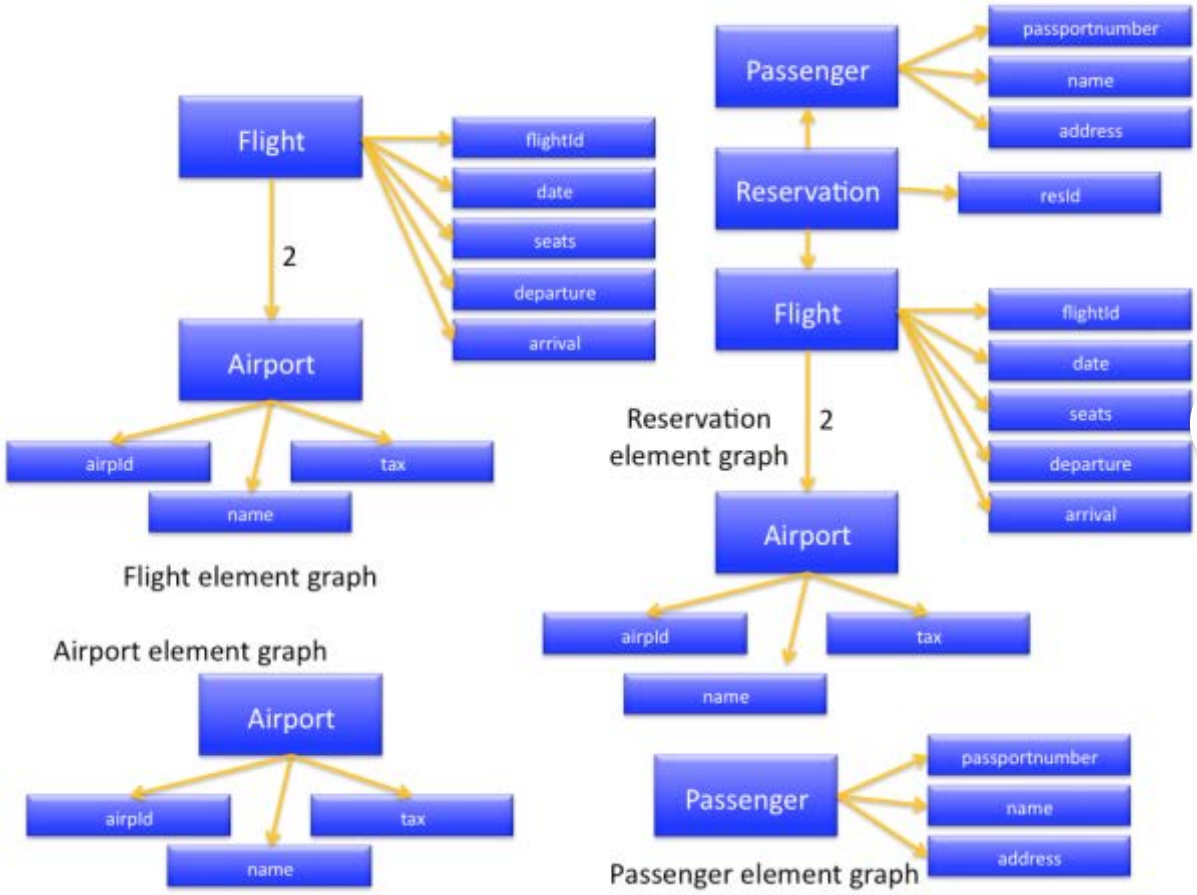
Schema-based Shredding: The DTD

```
<!ELEMENT Doc(Airport+, Passenger+, Flight+, Reservation+ >
<!ELEMENT Passenger(passportNumber, name, address) >
<!ELEMENT Airport(airpId, name, tax) >
<!ELEMENT Flight(flightId, date, departure, arrival, seats, Airport, Airport) >
<!ELEMENT Reservation(resId, Flight, Passenger)>
<!ELEMENT passportNumber #PCDATA>
<!ELEMENT name #PCDATA>
<!ELEMENT address #PCDATA>
<!ATTLIST Airport airpId CDATA>
<!ELEMENT tax #PCDATA>
<!ATTLIST Flight flightId CDATA>
<!ELEMENT date #PCDATA>
<!ELEMENT departure #PCDATA>
<!ELEMENT arrival #PCDATA>
<!ELEMENT seats #PCDATA>
<!ELEMENT resId #PCDATA>
```

Schema-based Shredding: The DTD graph



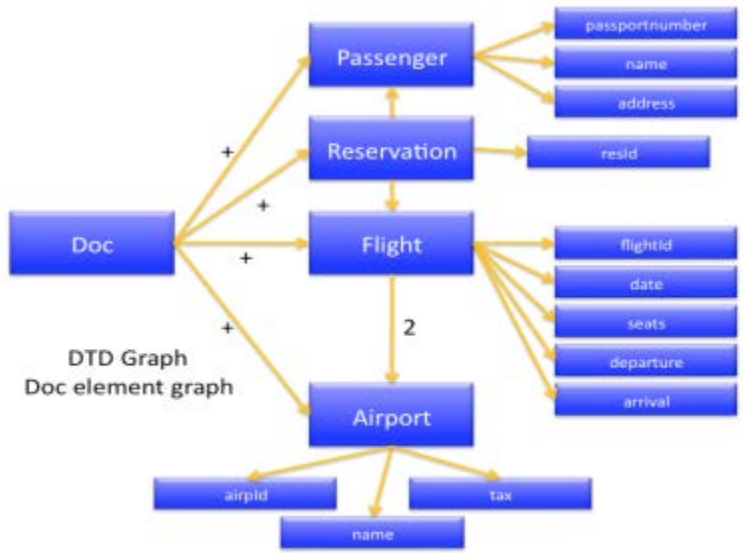
Schema-based Shredding: The element (sub)graphs



Since there are no cycles, these are all subgraphs.

Schema-based Shredding: DB Schema for the Doc subgraph

Doc(DocID)



We can not put the children inline because they are not unique...

Schema-based Shredding: DB Schema for the Doc subgraph

Doc(DocID)

Doc.Passenger(Doc.PassengerID, Doc.Passenger.parentID, Doc.Passenger.passportNumber, Doc.Passenger.name, Doc.Passenger.address)

Doc.Airport(Doc.Airport.airpId, Doc.Airport.parentID, Doc.Airport.name, Doc.Airport.address)

Doc.Flight(Doc.FlightID, Doc.Flight.flightID, Doc.Flight.date, Doc.Flight.seats, Doc.Flight.departure, Doc.Flight.arrival)

Doc.Reservation(Doc.ReservationID, Doc.Reservation.parentID, Doc.Reservation.resId, Doc.Reservation.Passenger.passportnumber, Doc.Reservation.Passenger.name, Doc.Reservation.Passenger.address, Doc.Reservation.Flight.flightID, Doc.Reservation.Flight.date, Doc.Reservation.Flight.seats, Doc.Reservation.Flight.departure, Doc.Reservation.Flight.arrival)

So we create special tables for them

Schema-based Shredding: DB Schema for the Doc subgraph

Doc(DocID)

Doc.Passenger(Doc.PassengerID, Doc.Passenger.parentID, Doc.Passenger.passportNumber, Doc.Passenger.name, Doc.Passenger.address)

Doc.Airport(Doc.Airport.airpId, Doc.Airport.parentID, Doc.Airport.name, Doc.Airport.address)

Doc.Flight(Doc.FlightID, Doc.Flight.flightID, Doc.Flight.date, Doc.Flight.seats)

Doc.Reservation(Doc.ReservationID, Doc.Reservation.parentID, Doc.Reservation.resId, Doc.Reservation.Passenger.passportnumber, Doc.Reservation.Passenger.name, Doc.Reservation.Passenger.address, Doc.Reservation.Flight.flightID, Doc.Reservation.Flight.date, Doc.Reservation.Flight.seats)

Here, there are attributes we can put inline.

Schema-based Shredding: DB Schema for the Doc subgraph

Doc(DocID)

Doc.Passenger(Doc.PassengerID, Doc.Passenger.parentID, Doc.Passenger.passportNumber, Doc.Passenger.name, Doc.Passenger.address)

Doc.Airport(Doc.Airport.airpId, Doc.Airport.parentID, Doc.Airport.name, Doc.Airport.address)

Doc.Flight(Doc.FlightID, Doc.Flight.flightID, Doc.Flight.date, Doc.Flight.seats)

Doc.Flight.Airport(Doc.Flight.AirportID, Doc.Flight.Airport.parentID, Doc.Flight.Airport.airpID, Doc.Flight.Airport.name, Doc.Flight.Airport.tax, position)

Doc.Reservation(Doc.ReservationID, Doc.Reservation.parentID, Doc.Reservation.resId, Doc.Reservation.Passenger.passportnumber, Doc.Reservation.Passenger.name, Doc.Reservation.Passenger.address, Doc.Reservation.Flight.flightID, Doc.Reservation.Flight.date, Doc.Reservation.Flight.seats)

Doc.Reservation.Flight.Airport(Doc.Reservation.Flight.AirportID, Doc.Reservation.Flight.Airport.parentID, Doc.Reservation.Flight.Airport.airpId, Doc.Reservation.Flight.Airport.name, Doc.Reservation.Flight.Airport.tax, position)

Those need a special table as well, because they can occur more than once.

Schema-based Shredding: The other subgraphs

Reservation(ReservationID, Reservation.parentID, Reservation.resId, Reservation.Passenger.passportnumber, Reservation.Passenger.name, Reservation.Passenger.address, Reservation.Flight.flightID, Reservation.Flight.date, Reservation.Flight.seats, Reservation.Flight.departure, Reservation.Flight.arrival)

Reservation.Flight.Airport(Reservation.Flight.AirportID, Reservation.Flight.Airport.parentID, Reservation.Flight.Airport.airpId, Reservation.Flight.Airport.name, Reservation.Flight.Airport.tax, position)

Flight(FlightID, Flight.flightID, Flight.date, Flight.seats, Flight.departure, Flight.arrival)

Flight.Airport(Flight.AirportID, Flight.Airport.parentID, Flight.Airport.airpID, Flight.Airport.name, Flight.Airport.tax, position)

Passenger(PassengerID, Passenger.parentID, Passenger.passportNumber, Passenger.name, Passenger.address)

Airport(AirportID, Airport.airpId, Airport.parentID, Airport.name, Airport.address)

name(nameID, name)
address(nameID, address)
...

Schema-based Shredding: Filling in the tables

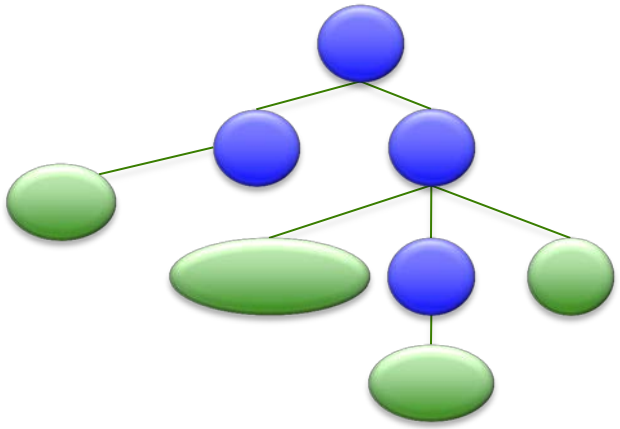
Doc.Flight.AirportID	Doc.Flight.Airport.parentID	Doc.Flight.Airport.airpId	Doc.Flight.Airport.name	Doc.Flight.Airport.tax	position
1	1	ZRH	Zurich	150	1
2	1	PAR	Paris	100	2
3	2	PAR	Paris	100	1
4	2	LHR	London Heathrow	100	2

Doc.Flight.Airport

Edge

Edge(Ordinal, Source, Label, Target)

ValueString(Id, Value)
ValueInteger(Id, Value)
ValueDate(Id, Value)
ValueTime(Id, Value)
ValueFloat(Id, Value)



Edge

Ordinal	Source	Label	Target
1	1	Doc	2
1	2	Airport	3
1	3	airId	v1
2	3	name	v2
3	3	tax	v3

```
<doc>
   <Airport airId="ZRH">
      <name>Zurich</name>
      <tax>150</tax>
   </Airport>
...
</doc>
```

Id	Value
v1	ZRH
v2	Zurich

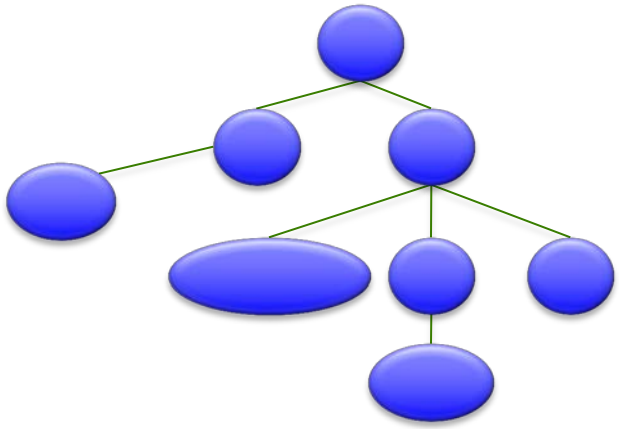
Id	Value
v3	150

Tree encoding

Tree encoding

Tree encoding: one single table

Tree(pre, size, level, kind, prop, frag)



Tree encoding

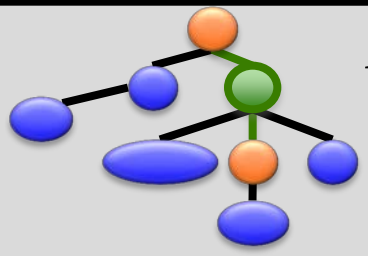
```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	ZRH	1
5	1	3	element	name	1
6	0	4	text	Zurich	

Tree encoding

```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	ZRH	1
5	1	3	element	name	1
6	0	4	text	Zurich	

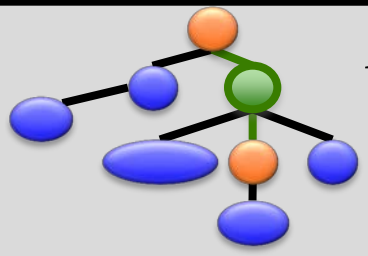


Ancestor relationship?

Tree encoding

```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	ZRH	1
5	1	3	element	name	1
6	0	4	text	Zurich	



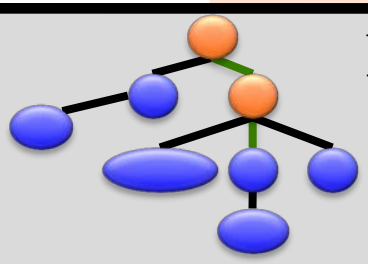
Ancestor relationship?

descendant.pre > ancestor.pre
& descendant.pre <= ancestor.pre+ancestor.size

Tree encoding

```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	ZRH	1
5	1	3	element	name	1
6	0	4	text	Zurich	1

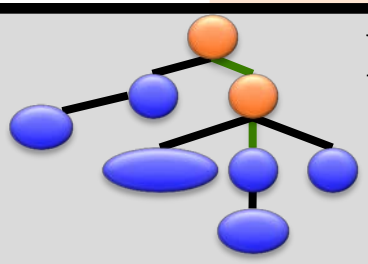


Parent relationship?

Tree encoding

```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	ZRH	1
5	1	3	element	name	1
6	0	4	text	Zurich	1



Parent relationship?

child.pre > parent.pre
& child.pre <= parent.pre+parent.size
& child.level = parent.level + 1

The problem we are solving today: Storing XML in an RDBMS

```
<root>Some <b>XML</b> document</root>
```

Relational Database

The problem we are solving today: Storing XML in an RDBMS

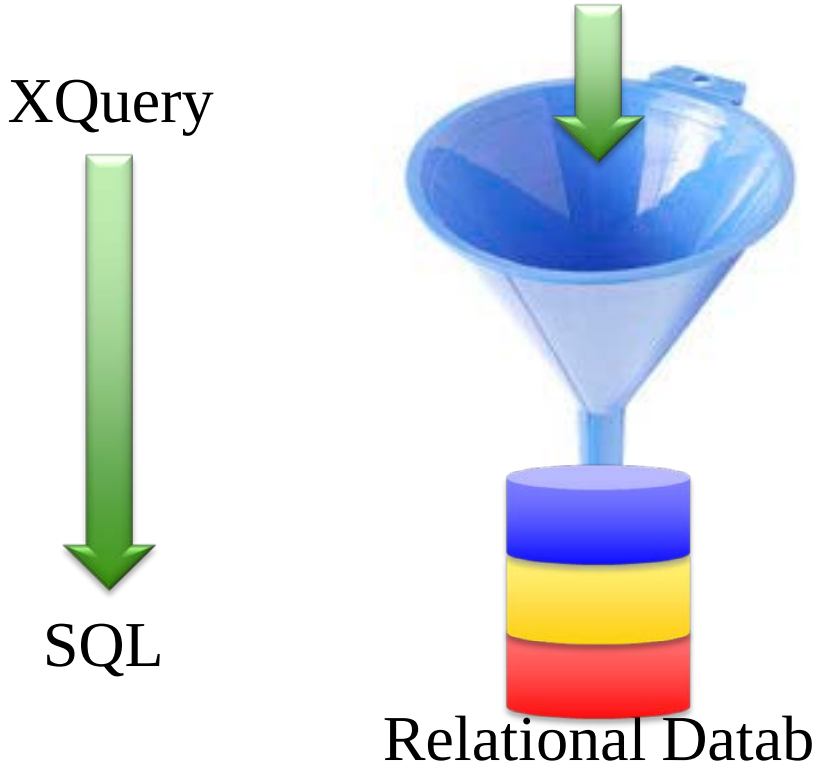
```
<root>Some <b>XML</b> document</root>
```

XQuery

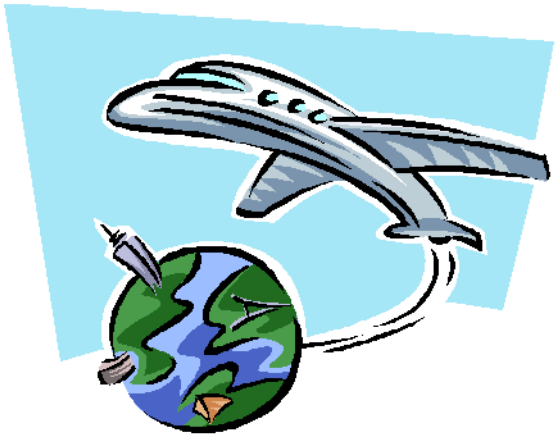
Relational Database

The problem we are solving today: Storing XML in an RDBMS

```
<root>Some <b>XML</b> document</root>
```



Our Query



"Determine all destinations of passenger Santa Claus"

XML Documents: Query for Schema-based Shredding (no nesting)

Passenger(passportNumber, name, address)
Airport(airpId, name, tax)
Flight(flightId, date, departure, arrival, seats, source, destination)
Reservation(flightRef, passportNumber)

XML Documents: Query for Schema-based Shredding (no nesting)

```
SELECT a.name FROM Reservation r, Flight f, Passenger p, Airport a
WHERE r.passRef = p.passportnumber
    AND p.name='Santa Claus'
    AND r.flightRef = f.flightId
    AND f.destination = a.airpId
```

XML Documents: Schema-based Shredding (nesting)

Doc(DocID)

Doc.Passenger(Doc.PassengerID, Doc.Passenger.parentID, Doc.Passenger.passportNumber, Doc.Passenger.name, Doc.Passenger.address)

Doc.Airport(Doc.Airport.airpId, Doc.Airport.parentID, Doc.Airport.name, Doc.Airport.address)

Doc.Flight(Doc.FlightID, Doc.Flight.flightID, Doc.Flight.date, Doc.Flight.seats, Doc.Flight.departure, Doc.Flight.arrival)

Doc.Flight.Airport(Doc.Flight.AirportID, Doc.Flight.Airport.parentID, Doc.Flight.Airport.airpID, Doc.Flight.Airport.name, Doc.Flight.Airport.tax, position)

Doc.Reservation(Doc.ReservationID, Doc.Reservation.parentID, Doc.Reservation.resId, Doc.Reservation.Passenger.passportnumber, Doc.Reservation.Passenger.name, Doc.Reservation.Passenger.address, Doc.Reservation.Flight.flightID, Doc.Reservation.Flight.date, Doc.Reservation.Flight.seats, Doc.Reservation.Flight.departure, Doc.Reservation.Flight.arrival)

Doc.Reservation.Flight.Airport(Doc.Reservation.Flight.AirportID, Doc.Reservation.Flight.Airport.parentID, Doc.Reservation.Flight.Airport.airpId, Doc.Reservation.Flight.Airport.name, Doc.Reservation.Flight.Airport.tax, position)

XML Documents: Query for Schema-based Shredding (nesting)

```
SELECT a.name from Doc.Reservation r, Doc.Reservation.Flight.Airport a
WHERE a.Doc.Reservation.Flight.Airport.parentID = r.Doc.ReservationID
    AND r.Doc.Reservation.Passenger.name="Santa Claus"
    AND a.position = "2"
```

Edge

Ordinal	Source	Label	Target
1	1	Doc	2
1	2	Airport	3
1	3	airId	v1
2	3	name	v2
3	3	tax	v3

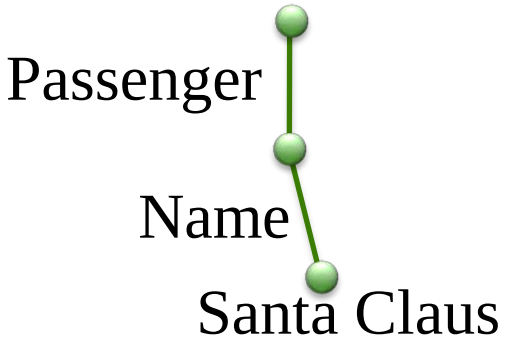
```
<doc>
   <Airport airId="ZRH">
      <name>Zurich</name>
      <tax>150</tax>
   </Airport>
...
</doc>
```

Id	Value
v1	ZRH
v2	Zurich

Id	Value
v3	150

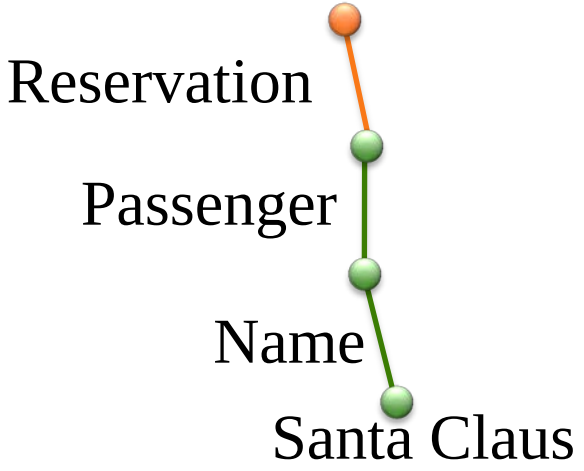
XML Documents: Query for Edge

```
Passenger_Vertices
SELECT P
FROM Edge P, Edge N, ValueString Vpname
WHERE P.Label = "Passenger"
AND P.Target = N.Source
AND N.Label = "name"
AND N.Target = Vpname.Id
AND Vpname.Value = “Santa Claus”
```



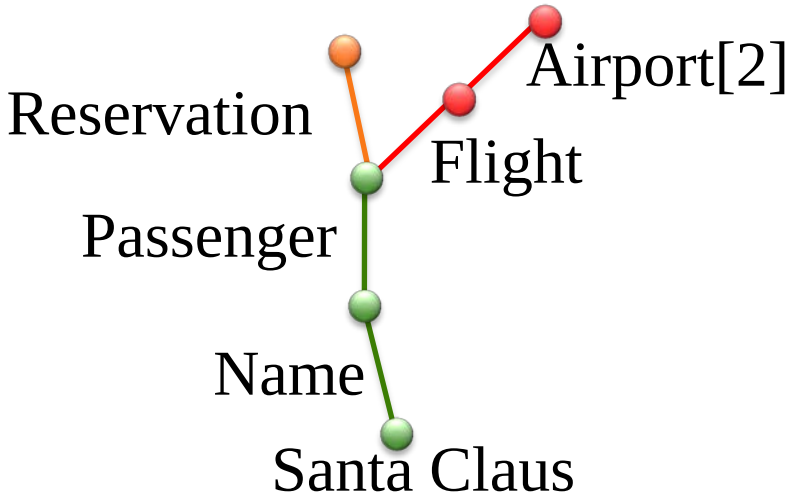
XML Documents: Query for Edge

```
Passenger_Vertices
SELECT P
FROM Edge P, Edge N, ValueString Vpname
WHERE P.Label = "Passenger"
AND P.Target = N.Source
AND N.Label = "name"
AND N.Target = Vpname.Id
AND Vpname.Value = “Santa Claus”

Reservations_Vertices
SELECT R
FROM Edge R, (Passenger_Vertices) P,
WHERE R.Name = "Reservation"
AND R.Target = P.Source
```



XML Documents: Query for Edge

```
Passenger_Vertices
SELECT P
FROM Edge P, Edge N, ValueString Vpname
WHERE P.Label = "Passenger"
AND P.Target = N.Source
AND N.Label = "name"
AND N.Target = Vpname.Id
AND Vpname.Value = “Santa Claus”

Reservations_Vertices
SELECT R
FROM Edge R, (Passenger_Vertices) P,
WHERE R.Name = "Reservation"
AND R.Target = P.Source

Destination_Vertices:
SELECT Vaname.Value
FROM Edge F,
     Edge G,
     (Reservation_Vertices) R,
     ValueString Vaname
WHERE F.Source = R.Target
AND F.Name = “Flight”
AND F.Target = G.Source
AND G.Name = "Airport"
AND G.Ordinal =  "2"
AND G.Target = Vaname.Id
```



XML Documents: Query for Edge

```
SELECT Vaname.Value
FROM Edge F,
     Edge G,
     (SELECT R
      FROM Edge R,
           (SELECT P
            FROM Edge P, Edge N, ValueString Vpname
            WHERE P.Label = "Passenger"
            AND P.Target = N.Source
            AND N.Label = "name"
            AND N.Target = Vpname.Id
            AND Vpname.Value = “Santa Claus”
           ),
      WHERE R.Name = "Reservation"
      AND R.Target = P.Source
     ),
     ValueString Vaname
WHERE F.Source = R.Target
AND F.Name = “Flight”
AND F.Target = G.Source
AND G.Name = "Airport"
AND G.Ordinal =  "2"
AND G.Target = Vaname.Id
```

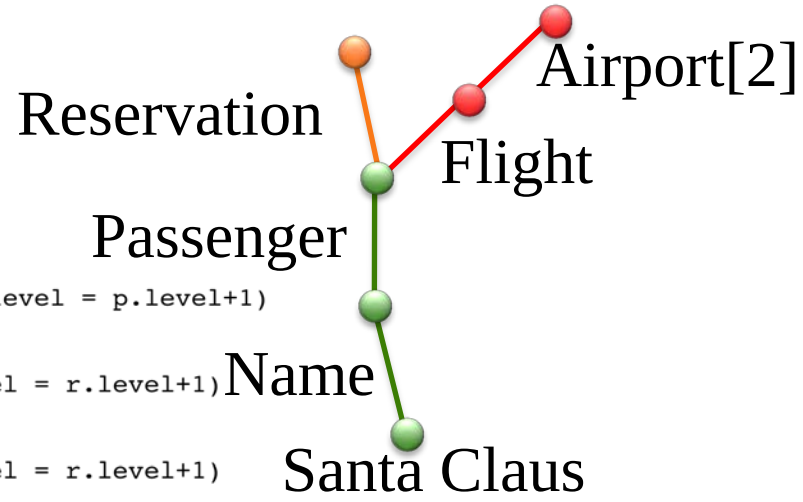
Tree encoding

```
<doc>
  <Airport airId="ZRH">
    <name>Zurich</name>
    <tax>150</tax>
  </Airport>
...
</doc>
```

pre	size	level	kind	prop	frag
1	(many)	1	element	Doc	1
2	6	2	element	Airport	1
3	1	3	attribute	airId	1
4	0	4	attvalue	LHR	1
5	1	3	element	name	1
6	0	4	text	London Heathrow	1

XML Documents: Query for Tree Encoding

```
SELECT dn.prop
FROM Tree p,
     Tree pn,
     Tree r,
     Tree f,
     Tree d,
     Tree dn
WHERE pn.prop="Santa Claus"
AND pn.kind="text"
AND (pn.pre > p.pre AND pn.pre <= p.pre + p.size AND pn.level = p.level+1)
AND p.prop="Passenger"
AND p.kind="element"
AND (p.pre > r.pre AND p.pre <= r.pre + r.size AND p.level = r.level+1)
AND r.prop="Reservation"
AND r.kind="element"
AND (f.pre > r.pre AND f.pre <= r.pre + r.size AND f.level = r.level+1)
AND f.prop="Flight"
AND f.kind="element"
AND (d.pre > f.pre AND d.pre <= f.pre + f.size AND d.level = f.level+1)
AND d.prop="Airport"
AND d.kind="element"
AND d.pre = (SELECT MAX(c.pre)
             FROM Tree c
             WHERE c.prop="Airport"
             AND c.kind="element"
             AND (c.pre > f.pre
                      AND c.pre <= f.pre + f.size and c.level = f.level+1)
            )
AND dn.kind="text
AND (dn.pre > d.pre AND dn.pre <= d.pre + d.size and dn.level = d.level+1)
```



XML Documents: Query for Tree Encoding

```
SELECT dn.prop
FROM Tree p,
     Tree pn,
     Tree r,
     Tree f,
     Tree d,
     Tree dn
WHERE pn.prop="Santa Claus"
AND pn.kind="text"
AND (pn.pre > p.pre AND pn.pre <= p.pre + p.size AND pn.level = p.level+1)
AND p.prop="Passenger"
AND p.kind="element"
AND (p.pre > r.pre AND p.pre <= r.pre + r.size AND p.level = r.level+1)
AND r.prop="Reservation"
AND r.kind="element"
AND (f.pre > r.pre AND f.pre <= r.pre + r.size AND f.level = r.level+1)
AND f.prop="Flight"
AND f.kind="element"
AND (d.pre > f.pre AND d.pre <= f.pre + f.size AND d.level = f.level+1)
AND d.prop="Airport"
AND d.kind="element"
AND d.pre = (SELECT MAX(c.pre)
             FROM Tree c
             WHERE c.prop="Airport"
             AND c.kind="element"
             AND (c.pre > f.pre
                      AND c.pre <= f.pre + f.size and c.level = f.level+1)
            )
AND dn.kind="text
AND (dn.pre > d.pre AND dn.pre <= d.pre + d.size and dn.level = d.level+1)
```

pn offspring of p

Good luck with the exam!