**Fall, 2005 CIS 550**

# Database and Information Systems

# Homework 5

November 15, 2005; Due November 22, 2005 at 1:30 pm

For this homework, you should test your answers using Galax, the same XQuery processor we used for Homework 4. You can ssh to **eniac-l.seas.upenn.edu** and run ~zives/galax/bin/galax-run on your query source file(s).

Consider how to *integrate* different instances of the PBAY auction system, in order to create a "unified PBAY." Two schemas, derived from student answers to the previous homework, will be the basis of this assignment. These schemas are available at ~zives/galax/schema-a.xsd and ~zives/galax/schema-b.xsd, with corresponding sample data sets ~zives/galax/data-a.xml and ~zives/galax/data-b.xml.

**Problem 1 [25 points]:** Write an XML Schema capturing an integrated view of the two schemas. The output schema contain the significant concepts from the two schemas, and it should be nested – with **items** in the outer level, and the associated **sellers** in the lower level.

**Problem 2 [25 points]:** Write two schema mappings (views) in XQuery, one over schema-a and the other over schema-b, that output XML conforming to your integrated schema in Problem 1. Finally, define an additional view that simply unions together output from the two views under a common root tag. This final view defines the contents of the *mediated schema* from the output of the schema mappings.

**Problem 3 [25 points]:** Write the following query in XQuery over your mediated schema view from the previous problem: Find all names of items sold by the seller with email mike@wharton.upenn.edu.

**Problem 4 [25 points]:** Manually write the *unfolding* of the previous query into a query directly over Schema $A$, i.e., merge the query and the mapping to schema-a, such that you have a query that is posed directly over schema-a.

**Problem 5 [Extra credit, 10 points]:** Suppose that the two schemas use **sellerID**s that are assigned using different schemes — some sellers in each schema are shared with the other schema, except that their IDs are different. Briefly explain how one would need to change the mappings (and what other data would be needed).

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://www.mystore.org" xmlns:st="http://www.mystore.org"
 targetNamespace="http://www.mystore.org" elementFormDefault="qualified">
<xs:element name="Store">
<xs:complexType>
<xs:sequence>
<xs:element name="Sellers" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="sellerID" type="xs:int"/>
<xs:element name="rating" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
<xs:element name="phone" type="xs:string"/>
<xs:element name="nickname" type="xs:string" minOccurs="0"/>
<xs:element name="membersince" type="xs:string"/>
<xs:element name="memberdue" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Items" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="itemID" type="xs:int"/>
<xs:element name="type" type="xs:string"/>
<xs:element name="itemname" type="xs:string"/>
<xs:element name="condition" type="xs:string"/>
<xs:element name="description" type="xs:string"/>
<xs:element name="manufacturer" type="xs:string" minOccurs="0"/>
<xs:element name="remark" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Stock" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="itemID" type="xs:int"/>
<xs:element name="sellerID" type="xs:int"/>
<xs:element name="sincedate" type="xs:string"/>
<xs:element name="duedate" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="PKSellers">
<xs:selector xpath="st:Sellers"/>
<xs:field xpath="./sellerID"/>
</xs:key>
<xs:key name="PKItems">
<xs:selector xpath="st:Items"/>
<xs:field xpath="./itemID"/>
</xs:key>
<xs:key name="PKStock">
<xs:selector xpath="st:Stock"/>
<xs:field xpath="./sellerID"/>
<xs:field xpath="./itemID"/>
</xs:key>
<xs:keyref name="RPKStock1" refer="PKSellers">
<xs:selector xpath="st:Stock"/>
<xs:field xpath="./sellerID"/>
</xs:keyref>
<xs:keyref name="RPKStock2" refer="PKItems">
<xs:selector xpath="st:Stock"/>
<xs:field xpath="./itemID"/>
</xs:keyref>
</xs:element>
</xs:schema>
```

Figure 1: Schema $A$

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://www.mystore.org" xmlns:st="http://www.mystore.org"
 targetNamespace="http://www.mystore.org" elementFormDefault="qualified">
   <xs:simpleType name="email">
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:complexType name="sellerType">
      <xs:sequence>
         <xs:element name="sellerEmail" type="email"/>
         <xs:element name="firstName" type="xs:string"/>
         <xs:element name="lastName" type="xs:string"/>
         <xs:element name="positiveFeedback" type="xs:int"/>
         <xs:element name="negativeFeedback" type="xs:int"/>
         <xs:element name="forSale" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="itemForSale" type="xs:int" nillable="false"/>
                  <xs:element name="quantity" type="xs:int" nillable="false"/>
                  <xs:element name="unitprice" type="xs:float"/>
                  <xs:element name="pictureLink" type="xs:string" minOccurs="0"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
      <xs:attribute name="sellerID" type="xs:int" use="required"/>
      <xs:attribute name="rating" use="required">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:minLength value="1"/>
               <xs:maxLength value="1"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:complexType name="itemType">
      <xs:sequence>
         <xs:element name="name" type="xs:string"/>
         <xs:element name="description" type="xs:string"/>
         <xs:element name="madeBy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="itemID" type="xs:int" use="required"/>
      <xs:attribute name="type" type="xs:string"/>
   </xs:complexType>
   <xs:element name="PBAY">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="seller" type="sellerType" maxOccurs="unbounded"/>
            <xs:element name="item" type="itemType" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
      <xs:key name="sellerKey">
         <xs:selector xpath="st:seller"/>
         <xs:field xpath="@sellerID"/>
      </xs:key>
      <xs:key name="itemKey">
         <xs:selector xpath="st:item"/>
         <xs:field xpath="@itemID"/>
      </xs:key>
      <xs:keyref name="SellerItemKey" refer="itemKey">
         <xs:selector xpath="st:seller/forSale"/>
         <xs:field xpath="./itemForSale"/>
      </xs:keyref>
   </xs:element>
</xs:schema>
```

Figure 2: Schema $B$