Assignment 3: XML and Database

Due date: 2nd, June
Submit a package of two files: Products.xml (1) and Queries.txt (2,3). Include your name and student ID.

Consider the following relational data:

pid	Name	Price	Description
323	gizmo	22.99	great
233	gizmo plus	99.99	more features
312	gadget	59.99	good value

Table 1: Products

sid	Name	Phone
s282	Wiz	555-1234
s521	Econo-Wiz	555-6543

Table 2: Stores

pid	Markup	sid
323	0%	s521
233	25%	s282
233	15%	s521

Table 3: Sells

1. **[3 marks]** We want to export this data into an XML file. Write a DTD describing the following structure for the XML file:

   - there is one root element called products
   - the products element contains a sequence of product sub elements, one for each product in the database
   - each product element contains one name, one price, and one description subelement, and a sequence of store subelements, one for each store that sells that product:
   - each store element contains one name, one phone, and one markup .

   Write the XML document obtained by exporting the database above; you have to turn in an XML document called Products.xml. Place the DTD on the top of Products.xml. Validate the xml file with xmllint.

2. **[3 marks]** Assuming that you have XML documents with the structure given in 1, write an XQuery that returns the names and prices of all products that are sold at least at one store with a markup of 25%. Write the same query in SQL over the original relational database schema. Turn in the two queries.

3. **[4 marks]** Assume the same database is represented in an XML document whose structure follows the relational tables:

```
<db>
        <products>
                <row>
                        <name> gizmo </name>
                        <price> 22.99 </price>
                        <description> great </description>
                <row>
        <row> ... <row>
        <row> ... <row>

        </products>

        <stores>
                <row> ... <row> ...
        </stores>

        <sells>
                <row> ... </row> ...
        </sells>
</db>
```

Write an XQuery that, when given an input with the structure described in 1), constructs an XML document with this structure. You have to turn in an XQuery.