Problem Session 4

Project 1 Help

Yusheng Yang
Stanford University
10/22/2007
Outline

- Announcements
- Synopsis of Project 1
- Parser demo
- Tips
- Q&A
Announcements

◆ Gradiance – two due Wed 10/24
◆ Challenge Problems #2 due Wed 10/24
◆ Project 1 due Wed 10/31
◆ Midterm Wed 10/31 11am-12:15pm Gates B01
  ♦ Up to and including 10/24 lecture on XPath
  ♦ Next problem session = review session
Synopsis of Project 1

Synopsis  Demo  Tips
3 Steps to Project 1

Plan your tables

Parse XML into tables

Prove everything worked
Plan Your Tables

Input: XML

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Bars SYSTEM "bars.dtd">
<Bars>
  <Bar>
    <Name>Joe's Bar</Name>
    <Beer Price="$2.50">
      <Name>Bud</Name>
    </Beer>
    <Beer Price="$3.00">
      <Name>Miller</Name>
    </Beer>
  </Bar>
</Bars>

Output: Table

<table>
<thead>
<tr>
<th>Sells</th>
<th>Bar</th>
<th>Beer</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe’s Bar</td>
<td>Bud</td>
<td>$2.50</td>
<td></td>
</tr>
<tr>
<td>Joe’s Bar</td>
<td>Miller</td>
<td>$3.00</td>
<td></td>
</tr>
</tbody>
</table>
Parse XML Into Tables

Input: XML

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Bars SYSTEM "bars.dtd">
<Bars>
  <Bar>
    <Name>Joe's Bar</Name>
    <Beer Price="$2.50">
      <Name>Bud</Name>
    </Beer>
    <Beer Price="$3.00">
      <Name>Miller</Name>
    </Beer>
  </Bar>
</Bars>

Output: data file for sqlldr

Joe's Bar<>Bud<>2.50
Joe's Bar<>Miller<>3.00

MyParser.java
Prove Everything Works

Just like Project 0

- Create table
- Run sqlldr
- Query, query, query!
Parser Demo

Synopsis

Demo

Tips
Element root = doc.getDocumentElement(); // get the root elem
Element[] bars = getElementByTagNameNR(root, "Bar"); // get the bar elems

/* Parse the Sells(bar, beer, price) relation */
for (int i = 0; i < bars.length; i++) { // loop through bars
    String barStr = getElementTextByTagNameNR(bars[i], "Name"); // get Sells.bar

    // get the current bar elem's beer elems
    Element[] beers = getElementByTagNameNR(bars[i], "Beer");

    for (int j = 0; j < beers.length; j++) ( // loop through beers
        String beerStr = getElementTextByTagNameNR(beers[j], "Name"); // get Sells.beer
        String priceStr = strip(beers[j].getAttribute("Price"));

        // this is how to print to Sells.dat
        streamSells.println(barStr + columnSeparator
                            + beerStr + columnSeparator
                            + priceStr);

        // this is how to print to screen
        /* System.out.println(barStr + columnSeparator
                            + beerStr + columnSeparator
                            + priceStr); */
    }
}

/******************************************************************************/

public static void main (String[] args) {
Tips

Synopsis

Demo

Tips
Tips

- [http://www.w3.org/2003/01/dom2-javadoc/org/w3c/dom/package-summary.html](http://www.w3.org/2003/01/dom2-javadoc/org/w3c/dom/package-summary.html)
- No separate user data, must infer from items (Users = Bidders + Sellers)
- “<>” is a good column separator.
- Make sure you read and understand items.txt and items.dtd
- Use DATE ‘MON-DD-RRRR HH24:MI:SS’ for dates in the control file
- Use NUMBER(8,2) for currency columns in CREATE TABLE.
Tips

• Some of the XML elements are optional (identify them in the DTD) and make sure you handle them with care in the parser. (Hint: if no such element exists, getElementByTagNameNR returns null)
• Run your parser on items-0.xml first, verify that it’s behaving correctly, then unleash it upon the rest of the data.
• strip() currency data in the parser.
What to Submit

• Your parser source
• Makefile for compiling your parser
• Your .ctl files
• create.sql, drop.sql, queries.sql
• runParser
• README.txt

• Create a .tar file containing the above.
• Please don’t submit any binaries or outputs.
Submission Rules

- runParser must ONLY have command(s) that run your parser and generate load files. It must NOT:
  - Drop or create tables
  - Load data into Oracle
  - Run any queries
- runLoad is for your own use, no need to submit it.
How We Test Your Submission

• For each submission, we will:
  – Run your drop.sql & create.sql on our own test database (we won’t touch your database)
  – Run your runParser
  – Load the generated load files into our database and ensure no errors occur
  – Run queries.sql on our database
• Repeat for next submission
Q & A

Q: In sqlplus, how do I display all of the tables I’ve created?
A: SELECT TABLE_NAME FROM USER_TABLES;

Questions?
First: http://www.stanford.edu/class/cs145/project1_faq.html
Second: Coursework Discussion
Last resort: cs145-aut0708-staff@lists.stanford.edu