CIS 560 Database System Concepts

Spring 2008

Homework 2 of 10: Machine Problem (MP2)

Implementing Relational Joins

Assigned: Sun 10 Feb 2008

Due: Wed 20 Feb 2008 (before midnight)

The purpose of this assignment is to exercise your basic understanding of relational algebra operations through implementation.

This homework assignment is worth a total of 20 points.

Use your KSOL drop box to turn in a.zip file MP2-XYZ.zip, with your initials in place of XYZ.

Use C++ or Java **only** to solve the first three problems. Specify which you are using in a README.txt file and name the programs accordingly: each problem should have a source code file mp2_i.[LANGUAGE]. Put the compile line in your README.txt file. The file names should be given on the command line, e.g., "mp2_i file1 file2". The file format consists of a list of attributes for the relation, each beginning with @, then zero or more rows.

For problem 1, refer to:

- International Business Machines, Inc. (2005). "Queries and the Query Optimizer: The Query Plan: Join Plan", *IBM Informix Dynamic Server Performance Guide*. Retrieved Sunday, 10 Feb 2008 from **http://snurl.com/1zfb1**
- Wikipedia. (2008). *Nested-Loop Join.* Retrieved Sunday, 10 February 2008 from **http://en.wikipedia.org/wiki/Nested_loop_join**

1. (6 points) Natural Join / Equality Join. Read `man join` on a Unix or Linux system such as `cislinux` and implement *equijoin* on two relations given as space-delimited files.

Note: The schemas of the relations r and s are assumed to share exactly one attribute, the *join field*. Your program should behave accordingly and place the join field in the first line of the schema header. Use `join` in Unix to check the output of your program. The commands for the relations below, stored in files `mp2-1-r.txt` and `mp2-1-s.txt`, are:

```
tail -6 mp2-1-r.txt | sort -k 3 > mp2-1-r_.txt
tail -2 mp2-1-s.txt | sort -k 2 > mp2-1-s_.txt
join -1 3 -2 1 mp2-1-r_.txt mp2-1-s_.txt > mp2-1-t_.txt
head -3 mp2-1-r_.txt > mp2-1-t.txt
head -3 mp2-1-s_.txt >> mp2-1-t.txt
cat mp2-1-t_.txt >> mp2-1-t.txt
rm mp2-1*_.txt
```

Turn in your source code (e.g., `mp2-1.java`) and a shell script `mp2-1.sh` containing all of the operations used to produce the target output for the example. Specify which shell you are using in a file README.txt or readme.pdf.

Example:

Relation r_1 (X1, X2, X3)

```
@X1
@X2
@X3
A1 B1 C1
A1 B2 C2
A2 B2 C2
A3 B2 C2
A4 B1 C3
A3 B1 C1
```

Relation s_1 (X3, X4)

```
@X3
@X4
C1 D1
C2 D2
```

Relation t_1 = r_1 s_1

```
@X3
@X1
@X2
@X4
C1 A1 B1 D1
C1 A3 B1 D1
C2 A1 B2 D2
C2 A2 B2 D2
C2 A3 B2 D2
```

For problems 2 – 3, refer to:

- Wikipedia. (2008). *Block Nested-Loop Join.* Retrieved Sunday, 10 February 2008 from **http://en.wikipedia.org/wiki/Block_nested_loop**

2. **(4 points) Block Nested-Loop Join.** Implement a *block nested-loop join*. Test it on the following relations:

Relation r_2 (X1, X2, X3, X4)

```
@X1
@X2
@X3
@X4
A, A2, A3, A4
C, C2, C3, C4
B, B2, B3, B4
C, Cx2, Cx3, Cx4
C, C2, C3, C4
B, B2, B3, B4
D, D2, D3, D4
```

Relation s_2 (X1, X5, X6)

@X1
@X5
@X6
A, A5, A6
B, B5, B6
C, C5, C6
E, E5, E6
B, B5, B6

Relation t_2 = r_2 s_2

@X1
@X2
@X3
@X4
@X5
@X6
A, A2, A3, A4 A5, A6
B, B2, B3, B4 B5, B6
B, B2, B3, B4 B5, B6
B, B2, B3, B4 B5, B6
B, B2, B3, B4 B5, B6
C, C2, C3, C4 C5, C6
C, C2, C3, C4 C5, C6
C, Cx2, Cx3, Cx4 C5, C6

Note: The same kind of preprocessing (header-trimming and sorting operations) should be performed as in MP2-1.

3. **(5 points) Schema handling.** Extend your program as shown in Lab 1 to actually handle the case of arbitrary schemas, which may have multiple shared attributes, and are not necessarily in order of attribute names. Test your solution with:

Relation s_3 (X4, X5, X2)

@X4
@X5
@X2
C4, E1, C2
C4, E2, C2
B4, E3, B2

4. **(5 points) Setting up Tomcat and JSP.** Following these preliminary instructions for MP4, MP6, and your term project should take you at most an hour or so. Notify the instructor (Dr. Hsu) if you run into any difficulties.

Last updated Sun 10 Feb 2008, based on an earlier version (April, 2006) by Tim Weninger.

a) Log into `cislinux.cis.ksu.edu` using SSH
b) Type: `% installtomcat5`
c) Type: Ctrl-Z (to get back to your shell, e.g., `bash`)
d) Type: `% java-config -s sun-jdk-1.5`
e) Type: `% java –version`

You should see:

```
java version "1.5.0_13"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_13-b05)
Java HotSpot(TM) Client VM (build 1.5.0_13-b05, mixed mode)
```

f) Type: `% echo "source ${HOME}/.gentoo/java-config-2" >> ~/.bashrc`
g) Type: `% source ~/.gentoo/java-config-2`
h) Type: `% cd tomcat5-linux/logs`
i) Type: `% ../bin/startup.sh`

You should see something like this:

```
Using CATALINA_BASE:   /home/faculty/bhsu/tomcat5-linux
Using CATALINA_HOME:   /home/faculty/bhsu/tomcat5-linux
Using CATALINA_TMPDIR: /home/faculty/bhsu/tomcat5-linux/temp
Using JRE_HOME:       /home/faculty/bhsu/.gentoo/java-config-2/current-
user-vm
```

and should have a file `catalina.out` in your logs directory.

j) Type: `% getent passwd <username>`
This shows `<username>:x:<userid>:<groupid>:Name:etc`

```
e.g., bhsu:x:2668:2668:Bill Hsu:/home/faculty/bhsu:/bin/tcsh
```

k) Open your web browser.
l) Go to http://cislinux.cis.ksu.edu:1[UID] (notice the "1" in front of your userid; the sysadmins wanted to make the port #'s >10000).

For example, uid 2668 will have web address
http://cislinux.cis.ksu.edu:12668/

m) JSP pages can be uploaded into the webapps/ROOT/, or you can create the page using a text editor such as vim or emacs. For instance: `vim test.jsp`

n) Type:

```
<html>
%@page import="java.util.Date"%
<h1>Today is <%=new Date()%></h1>
</html>
```

o) Quit out of your editor (ESC, :wq saves and exits vim)
p) http://cislinux.cis.ksu.edu:1[UID]/test.jsp
q) Make sure to run `% ../bin/shutdown.sh` when you're done.

Turn in a screenshot `mp2-4.jpg` of the JSP servlet loaded on your own page in a browser.

For the extra credit problem, consult the JavaDoc specifications for `currentTimeMillis()`:

- Sum Microsystems, Inc. (2004). “System”, *Java 2 Platform SE 5.0*. Retrieved Sunday, 10 Feb 2008 from **http://java.sun.com/j2se/1.5.0/docs/api/java/lang/System.html**

Extra Credit (2 points): Comparison of running times. Use gprof (in C/C++) or `currentTimeMillis` (in Java) to compare the running times of the implementations in MP2-1 and MP2-2.