CPS216 Advanced Database Systems (Data-Intensive Computing Systems) - Fall 2009
Assignment 1

- Total points = 80. Due date: Friday, Sept. 18, 2009 (5.00 PM).
- Submission: In class, or email solutions in pdf or plain text to shivnath@cs.duke.edu. You can also drop off the solutions at Shivnath's office: D338 LSRC.
- Do not forget to indicate your name on your submission.
- State all assumptions. For questions where descriptive solutions are required, you will be graded both on the correctness and clarity of your reasoning.
- Email questions to shivnath@cs.duke.edu.

Question 1 **Points 20 = 4 + 4 + 6 + 6**

Give two logical plans and two physical plans for the following SQL query.

```
Select R.A, T.B
From   R, S, T
Where R.A = S.A and R.B = T.B
```

Let the two physical plans you give be denoted $Plan_1$ and $Plan_2$. Describe a scenario where $Plan_1$ is better than $Plan_2$, and another where $Plan_2$ is better than $Plan_1$. For this question assume that the cost measure is the number of getNext() calls. (A complete example scenario may need to specify sizes of the tables, selectivity of joins, etc.)

Question 2 **Points 20 = 6 + 6 + 8**

Let $R_1(A, B)$ and $R_2(B, C)$ be two tables of data.

1. Suppose, neither R_1 nor R_2 has duplicate tuples. (That is, there is no pair of distinct tuples $r \in R_1$ and $s \in R_1$ such that $r.A = s.A$ and $r.B = s.B$; similarly for R_2.) What is the necessary and sufficient condition for the following equivalence to hold: $\sigma_{P_1 \bigvee P_2}(R_1 \bowtie R_2) = (\sigma_{P_1} R_1 \bowtie R_2) \bigcup_B (R_1 \bowtie \sigma_{P_2} R_2)$? Here, P_1 is a predicate that involves attributes in R_1 only, and P_2 is a predicate that involves attributes in R_2 only. $\bigcup_B$ denotes *bag union* (see Figure 1 for an illustration of bag and set unions.) State your condition as an expression in relational algebra. You may use ϕ to denote an empty set of tuples (i.e., a *null set*).
2. How does your answer to (1) change if R_1 and R_2 can have duplicate tuples in them? (That is, now there can be pairs of distinct tuples $r \in R_1$ and $s \in R_1$ such that $r.A = s.A$ and $r.B = s.B$; similarly for R_2.)
3. Does the following condition hold if R_1 and R_2 can have duplicate tuples in them: $\sigma_{P_1 \bigvee P_2}(R_1 \bowtie R_2) = (\sigma_{P_1} R_1 \bowtie R_2) \bigcup_S (R_1 \bowtie \sigma_{P_2} R_2)$? $\bigcup_S$ denotes *set union* (also called duplicate-eliminating union; see Figure 1.) If not, can you suggest a modification to right hand side of this condition so that the new condition holds? You can express your new condition in relational algebra or describe it in English.

$$\begin{array}{|c|c|}\hline 1 & a \\ \hline 2 & b \\ \hline\end{array} \cup_B \begin{array}{|c|c|}\hline 1 & a \\ \hline 3 & c \\ \hline\end{array} = \begin{array}{|c|c|}\hline 1 & a \\ \hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline 1 & a \\ \hline 2 & b \\ \hline\end{array} \cup_S \begin{array}{|c|c|}\hline 1 & a \\ \hline 3 & c \\ \hline\end{array} = \begin{array}{|c|c|}\hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline 1 & a \\ \hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline\end{array} \cup_B \begin{array}{|c|c|}\hline 1 & a \\ \hline 3 & c \\ \hline\end{array} = \begin{array}{|c|c|}\hline 1 & a \\ \hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline 1 & a \\ \hline 3 & c \\ \hline\end{array}$$

$$\begin{array}{|c|c|}\hline 1 & a \\ \hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline\end{array} \cup_S \begin{array}{|c|c|}\hline 1 & a \\ \hline 3 & c \\ \hline\end{array} = \begin{array}{|c|c|}\hline 1 & a \\ \hline 2 & b \\ \hline 3 & c \\ \hline\end{array}$$

Figure 1: Examples to illustrate Bag Union ($\bigcup_B$) and Set Union ($\bigcup_S$)

Question 3 **Points 20 = 10 + 10**

Figures 2(a)-(c) show three logical plans for the following SQL query over tables $R(A, B)$ and $S(A, C)$.

```
Select Distinct R.A
From   R, S
Where R.A = S.A
```

Note that "Select Distinct" in SQL represents a duplicate-eliminating projection. The logical operator $\pi_{R.A}$ in Figure 2 represents a duplicate-eliminating projection of attribute $R.A$, $\pi^p_{R.A}$ represents a duplicate-preserving projection of attribute $R.A$, and $\bowtie$ represents a natural join.

If we have a table with $R(A, B)$ tuples $\{\{1, a\}, \{1, b\}, \{2, c\}, \{3, d\}, \{4, e\}, \{4, f\}\}$, then a duplicate-preserving projection on $R.A$ will return $\{1, 1, 2, 3, 4, 4\}$, while a duplicate-eliminating projection on $R.A$ will return $\{1, 2, 3, 4\}$.

Assume that, in the general case, both R and S can contain duplicate tuples. Answer the following:

1. Is the logical plan in Figure 2(a) equivalent to the logical plan in Figure 2(b)? If not, what properties should the tables satisfy such that these plans are equivalent?

2. Is the logical plan in Figure 2(a) equivalent to the logical plan in Figure 2(c)? If not, what properties should the tables satisfy such that these plans are equivalent?

Question 4 **Points 20 = 8 + 12**

Figure 3 shows a physical execution plan for a query that joins four tables $R(A)$, $S(A)$, $T(B)$, and $U(B)$. Also shown are the tuples in the respective tables. TNLJ denotes the tuple nested loop

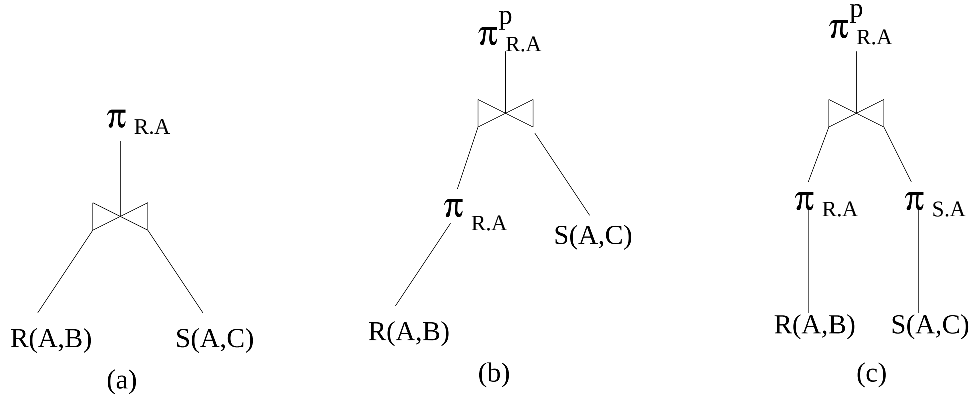

Figure 2: Logical execution plans for $\pi_A(R(A,B) \bowtie S(A,C))$

join that we discussed in class. Also, TableScan denotes a full scan of the table as we discussed in class.

1. Count the number of getNext() calls that the plan in Figure 3 will make. EOT (End-Of-Tuple) calls should be included in your answer.
2. Give the execution plan that will generate the minimum number of getNext() calls for this query. The plan you give can include TNLJ and TableScan operators only.

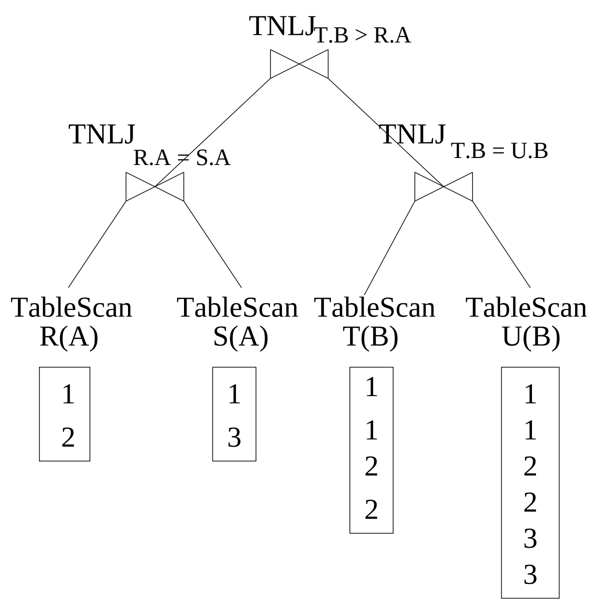

Figure 3: Physical execution plan for joining R, S, T, and U