CPS216 Advanced Database Systems - Fall 2007
Assignment 1

- Due date: Thursday, Sept. 13, 2007 (11.59 PM). Late submissions will not be accepted.
- Submission: In class, or email solution in pdf or plain text to shivnath@cs.duke.edu.
- 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* (also called *multiset union.*) 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.) If not, can you suggest a modification to the right side of this condition so that the new condition holds? (Hint: You may have to use one of the set difference or bag difference operators, denoted $-_S$ and $-_B$ respectively.)

Question 3

Points 20 = 10 + 10

Figures 1(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 1 represents a duplicate-eliminating projection of attribute $R.A$, and $\bowtie$ represents a natural join. Assume that, in the general case, both R and S can contain duplicate tuples.

1. Is the logical plan in Figure 1(a) equivalent to the logical plan in Figure 1(b)? If not, what is the minimal set of constraints on the tables such that these plans are equivalent?

2. Is the logical plan in Figure 1(a) equivalent to the logical plan in Figure 1(c)? If not, what is the minimal set of constraints on the tables such that these plans are equivalent?

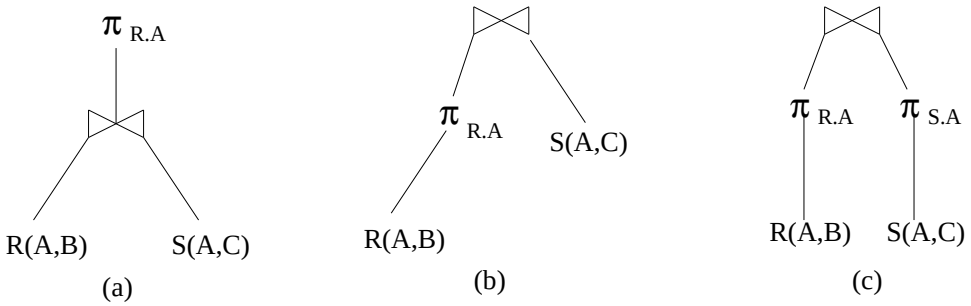

Figure 1: Logical execution plans for $\pi_A(R(A, B) \bowtie S(A, C))$

Question 4

Points 20 = 8 + 12

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

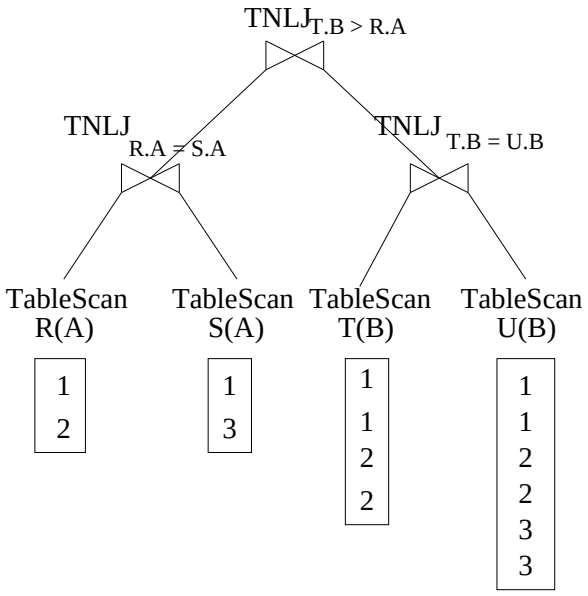

Figure 2: Physical execution plan for joining R, S, T, and U

Question 5

Points 10 = 2.5 + 2.5 + 2.5 + 2.5

Consider a 3.5 inch disk with 2 magnetic surfaces with 64 tracks per surface, rotating at 3600 rpm. It has a usable capacity of 2 megabytes (2×2^{20} bytes). Assume 20% of each track is used as overhead (gaps). Also, assume that the usable capacity is equally distributed among the tracks.

a. What is the burst bandwidth this disk can support?

b. What is the sustained bandwidth this disk can support?

c. What is the average rotational latency?

d. Assuming the average seek time is 16 ms, what is the average time to fetch a 2-kilobyte (2×2^{10} bytes) sector?

Question 6

Points 10 = 6 + 4

Consider a disk with the following properties:

- There are four platters providing eight surfaces.
- There are $2^{13} = 8192$ tracks per surface.
- There are (on average) $2^8 = 256$ sectors per track.
- There are $2^9 = 512$ bytes per sector.
- The disk rotates at 3840 rpm.
- The block size is $2^{12} = 4096$ bytes.
- Assume 10% of each track is used as overhead.
- The time it takes the head to move n tracks is 1 + n/500 milliseconds.

Suppose that we know that the last I/O request accessed cylinder 3000. (Cylinders are numbered sequentially: $1, 2, \ldots, 8192$.)

a. What is the expected (average) number of cylinders that will be traveled due to the very next I/O request to this disk?

b. What is the expected block access time for the next I/O, again given that the head is on cylinder 3000 initially?