# Data-Intensive Computing Systems

---

- Total points = 100.
- State all assumptions. For questions where descriptive solutions are required, you will be graded both on the correctness and clarity of your reasoning.

---

## Question 1 **Points 15**

This question is based on the following SQL query over tables R(A,B) and S(A,B).

```
Select    R.A, MAX(S.B)
From      R, S
Where     R.A = S.A and R.B < 20
Group By  R.A
```

Explain how MapReduce can be used **most efficiently** to process this query. That is, explain how many MapReduce jobs are needed, and what the respective Map phase(s), Reduce phase(s), and Combine phase(s) will do. Keep your answer brief and to the point. A couple of sentences are enough to convey the main ideas.

## Question 2 **Points 20**

Consider the following SQL query over tables $R(A)$, $S(A)$, and $T(A)$. Note that "Select Distinct" in SQL represents a duplicate-eliminating projection.

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

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 $\{1, 2, 3, 4\}$.

Figures 1(a)-(e) show five logical plans. The logical operator $\pi$ in Figure 1 represents a duplicate-eliminating projection. For example, $\pi_{R.A}$ represents a duplicate-eliminating projection of attribute $R.A$. On the other hand, the logical operator $\tau$ represents a duplicate-preserving projection. The logical operator $\bowtie$ represents a natural join. Assume that, in the general case, $R$, $S$, and $T$ can contain duplicate tuples.

1. Is the logical plan in Figure 1(a) equivalent to the logical plan in Figure 1(b)? (Equivalent means that they produce the same query result.) If not, what is the minimal set of properties 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 properties on the tables such that these plans are equivalent?
3. Is the logical plan in Figure 1(a) equivalent to the logical plan in Figure 1(d)? If not, what is the minimal set of properties on the tables such that these plans are equivalent?
4. Is the logical plan in Figure 1(a) equivalent to the logical plan in Figure 1(e)? If not, what is the minimal set of properties on the tables such that these plans are equivalent?

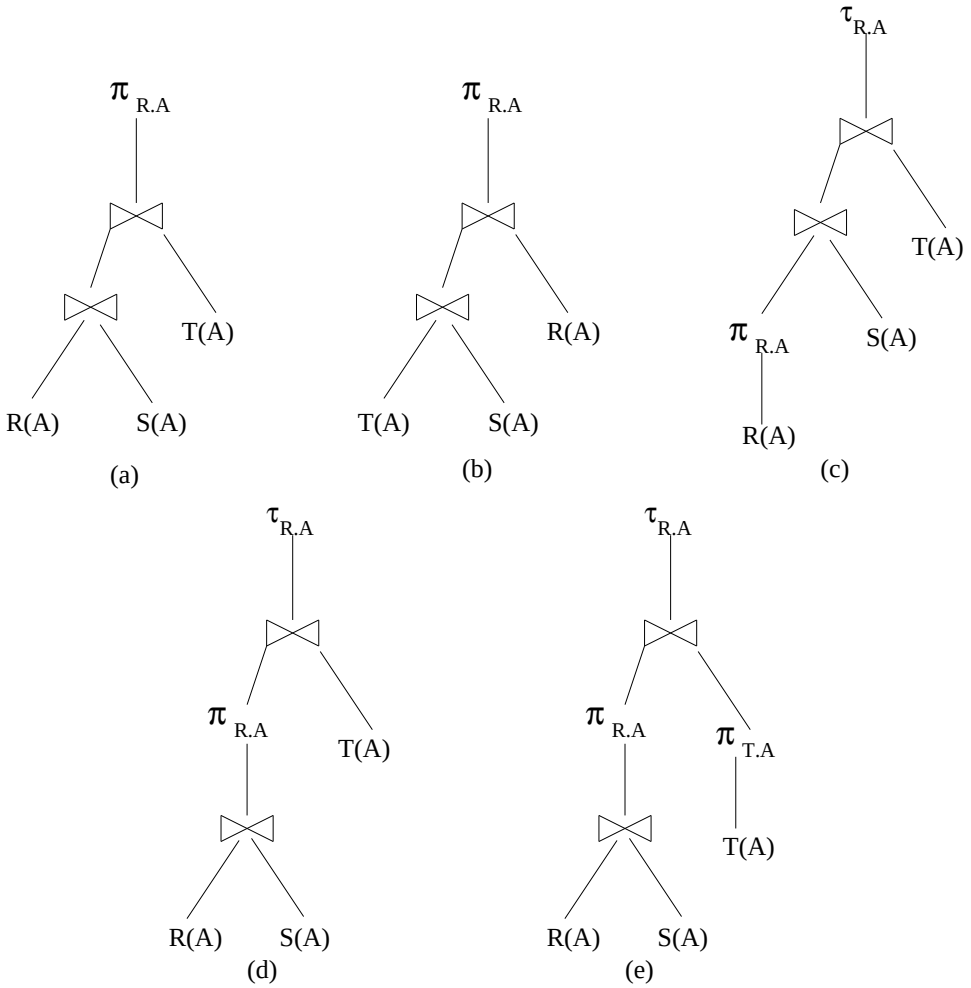


Figure 1: Logical plans

## Question 3 **Points 20**

In this question we will find the cost of physical plans in terms of the number of getNext() calls. Figure 2 shows the physical plan generated from each logical plan in Figure 1. We make the following assumptions:

A1 Data from each table is read through a table scan which is represented as *Scan*. (Note: Slide 43 of Notes 5 shows the iterator-based implementation of table scan. We do not consider index scans in this question.)

A2 Each join is implemented using a tuple nested loop join, or *TNLJ*. TNLJ is the only physical join operator available.

A3 A duplicate-eliminating projection is always implemented using the operator $\pi$. Note that $\pi$ is a physical operator here.

A4 A duplicate-preserving projection is always implemented using the operator $\tau$. Note that $\tau$ is a physical operator here.

The records in the tables are as follows.

1. $R(A)$ contains $\{1, 1, 2, 2, 3, 3\}$.

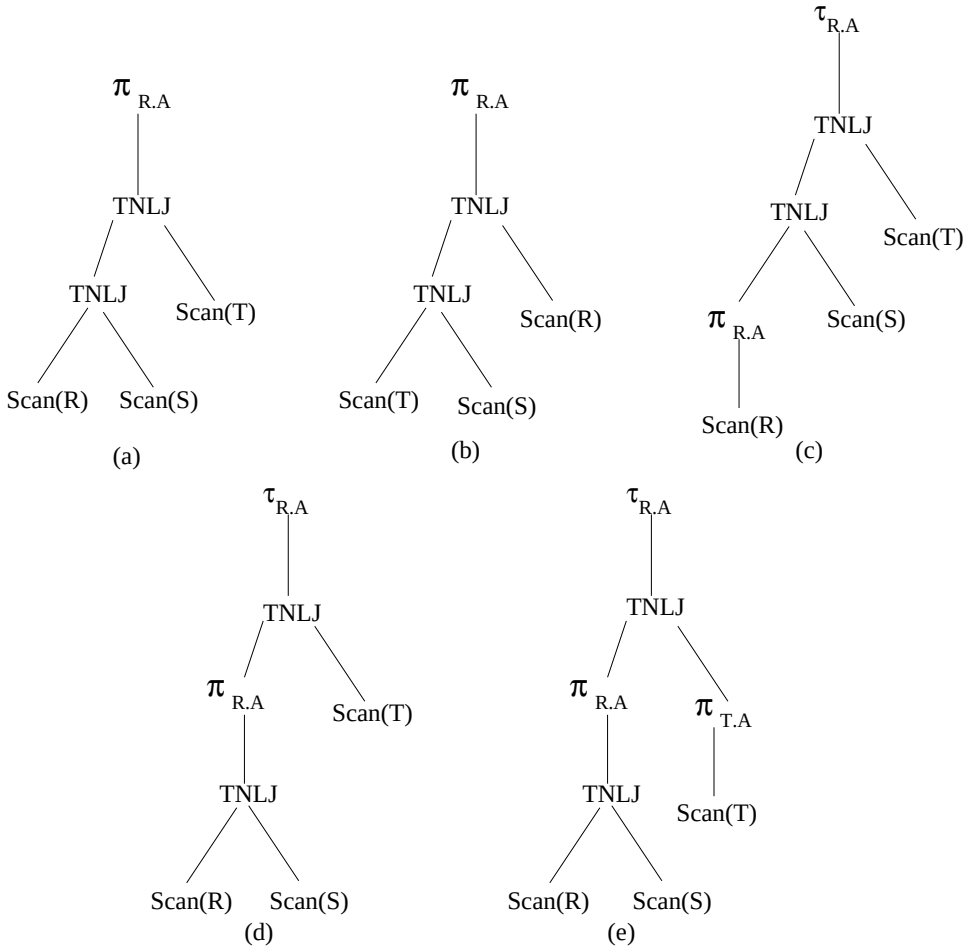


Figure 2: Physical plans

2. $S(A)$ contains $\{1, 2\}$.

3. $T(A)$ contains $\{1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4\}$.

Answer the following five questions based on the above information. Give explanations for partial credit.

1. How many getNext() calls are made by the plan in Figure 2(a)?

2. How many getNext() calls are made by the plan in Figure 2(b)?

3. How many getNext() calls are made by the plan in Figure 2(c)?

4. How many getNext() calls are made by the plan in Figure 2(d)?

5. How many getNext() calls are made by the plan in Figure 2(e)?

## Question 4 **Points 15**

Given the assumptions A1 to A4 in Question 3 about physical plans, find the number of possible physical plans for the SQL query in Question 2. Assume that $R$, $S$, and $T$ contain duplicate tuples. Give explanations for partial credit.

## Question 5 **Points 15**

Given the assumptions A1 to A4 in Question 3 about physical plans, find the physical plan that does the minimum number of getNext() calls for the SQL query in Question 2 and the records in tables $R(A)$, $S(A)$, and $T(A)$ given in Question 3. Give explanations for partial credit.

*Hint: The physical plan that does the minimum number of getNext() calls may not be one of the plans in Figure 2.*

## Question 6 **Points 15**

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 3 for an illustration of bag and set unions.) State your condition as an expression in relational algebra. You may use $\phi$ 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 3.) If not, can you suggest a modification to the 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.

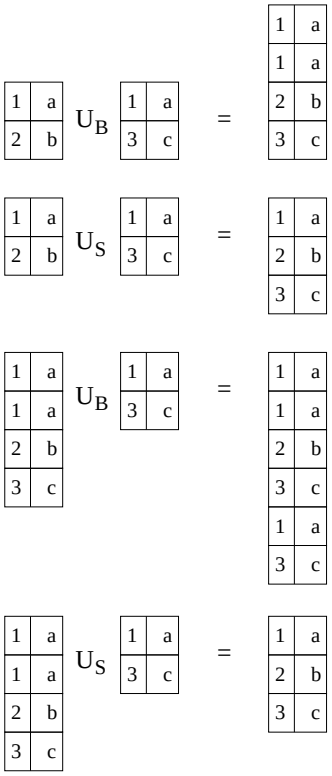


Figure 3: Examples to illustrate Bag Union ($\bigcup_B$) and Set Union ($\bigcup_S$)