## CS 317 - Data Management and I nformation Systems

## Professor Yan Chen

Homework 3 Solutions
Grader: Jin Hu

## Problem 5.2.1 and 5.2.2 (30 points)

Part b) (5 points)
Expression: $\quad \pi_{\text {maker }}\left(\sigma_{h d \geq 1}(\right.$ Product $\triangleright \triangleleft$ Laptop $\left.)\right)$
Results:
maker
--------
A
B
C
D
E

Part c) (5 points)

$$
\begin{aligned}
& A=\pi_{\text {model }}\left(\sigma_{\text {maker }=B}(\text { Product })\right) \\
& B=\pi_{\text {model, price }}(A \triangleright \triangleleft \text { PC }) \\
& C=\pi_{\text {model, price }}(A \triangleright \triangleleft \text { Laptop }) \\
& D=\pi_{\text {model, price }}(A \triangleright \triangleleft \text { Printer })
\end{aligned}
$$

Expression: $\quad B \cup C \cup D$

Results:

## model price

1004
1005
1006
2001
2002
2003
999
1499
2119
1448
2584
2738

Expression Tree:


## Part d) (5 points)

Expression: $\quad \pi_{\text {model }}\left(\sigma_{\text {color }=\text { true AND type }=\text { laser }}(\right.$ Printer $\left.)\right)$

Results:
Expression Tree:


Part e) (5 points)
Expression: $\quad \pi_{\text {maker }}\left(\sigma_{\text {type }=\text { laptop }}(\right.$ Product $\left.)\right)-\pi_{\text {maker }}\left(\sigma_{\text {type }=\text { PC }}(\right.$ Product $\left.)\right)$
Results:
Expression Tree:
mode
$\varnothing$


Part f) (5 points)
$\mathrm{PC}_{1}=\rho_{\mathrm{PC}_{1}}(\mathrm{PC})$
$\mathrm{PC}_{2}=\rho_{\mathrm{PC}_{2}}(\mathrm{PC})$

Expression: $\quad \pi_{\mathrm{PC}_{1} \text {.hd }}\left(\sigma_{\mathrm{PC}_{1} \text {.hd }=\mathrm{PC}_{2} \text {.hd AND PC } C_{1} \text {.model } \neq \mathrm{PC}_{2} \text {.model }}\left(\mathrm{PC}_{1} \times \mathrm{PC}_{2}\right)\right)$

Results:
Expression Tree:


Part g) (5 points)
$\mathrm{PC}_{1}=\rho_{\mathrm{PC}_{1}}(\mathrm{PC})$
$\mathrm{PC}_{2}=\rho_{\mathrm{PC}_{2}}(\mathrm{PC})$
Expression: $\quad \pi_{\mathrm{PC}_{1} \text {.model }} \mathrm{PC}_{2}$.model $\left(\sigma_{\mathrm{PC}_{1} \text {.speed }=}=\mathrm{PC}_{2}\right.$.speed AND PC $\mathrm{P}_{1}$ ram $=\mathrm{PC}_{2}$.ram AND $\mathrm{PC}_{1}$.model $<\mathrm{PC}_{2}$.model $\left.\left(\mathrm{PC}_{1} \times \mathrm{PC}_{2}\right)\right)$

Results:


Expression Tree:


## Problem 5.2.9 (10 points)

Part b) (5 points)

- Minimum:

0
$R$ and $S$ can have no tuples in common, yielding a NULL set.

- Maximum: mxn

Every tuple in R and S has the same common element. Thus, every tuple in R must join with every tuple in $S$.

Part c) (5 points)

- Minimum: 0

Condition C can be such that no tuple in R satisfy it. Thus, we have a NULL set.

- Maximum: mxn

Condition C can also be such that every tuple in R satisfy it. Thus, we have a full table crossed.

As set:
hd


10
60
20
40
80
30
7
As bag:
hd
-----
10
60
20
10
20
40
80
30
80
30
60
70
6

$$
\text { AVG(hd) }=35.3
$$

$$
\operatorname{AVG}(\mathrm{hd})=39
$$

## Problem 5.3.4 (20 points)

## Part b) (5 points)

Set: $\quad$ The $R \cap S$ results in tuples that belong to both $R$ and $S$. Thus, a tuple must appear in both $R$ and $S$ for it to appear in the intersection. If the operation is cascaded, then the result will consist of tuples that appear in all included relations. Thus, the result is the same, regardless of what order the operations occur in. If a tuple does not belong, it will eventually become eliminated.

Bag: Once the intersection operation finds a match, that pair of tuples is taken out of both relations. If there happens to be a second identical pair of tuples, it will be taken out again. Thus, no information is lost and the operation applies for bags.

Part c) (5 points)
Set: Natural join only looks at attributes, not tuples. This means that it will only merge the attributes together from the different relations, regardless of whether or not duplicate tuples exist. By the associative property of the operation, it does not matter what order the operation is performed. Because the operation does not look at tuples, it holds for sets.

Bag: Because natural join only looks at the attributes and not tuples, it does not distinguish between duplicate and unique tuples, the operation holds for sets.

Part g) (5 points)
Set: The projection operator only looks at attributes, not tuples. Thus, by similar reasoning from above, the projection operation will hold for sets. The union operation will look at the common elements between two relations. Thus, if a tuple appear in one relation, it will appear in the result. Thus, no information is lost and no duplication elimination was performed. For sets, there is no difference what order the operations are done. Thus, this operation holds for sets.

Bag: By the same process as above, this operation will hold for bags. When we project, however, we do not eliminate any elements. Thus, the union of two bags will be the same, regardless of when we project.

## Part i) (5 points)

Set: The selection operator looks at one tuple at a time - it does not distinguish between duplicate and unique ones. In ordered to be selected, a tuple in R must meet both conditions. Thus, if we select separately ( $C$ in one group and $D$ in the other), we know that a tuple must be in both groups to be part of the result. Thus, if we take the intersection of the two groups, we have our result. This is the same as imposing both constraints at the beginning.

Bag: By similar reasoning, if a duplicate tuple meets both conditions, then all other instances of that tuple will also meet both conditions. Thus, no duplicate tuples will be lost.

## Problem 5.3.5 (10 points)

## Part b) (5 points)

Set: To be part of the final result, a tuple must be common to either S or T and must be in $R$. The LHS checks the combined $S$ and $T$ relations against $R$ first whilst the RHS checks $S$ and $T$ separately against $R$ first before joining together. Both sides will yield the same result.

Bag: Counterexample

$$
\begin{aligned}
& R \cap(S \cup T)=(R \cap S) \cup(R \cap T) \\
\rightarrow & \{1\} \cap(\{1\} \cup\{1\})=(\{1\} \cap\{1\}) \cup(\{1\} \cap\{1\}) \\
\rightarrow & \{1\} \cap\{1\}=\{1\} \cup\{1\} \\
\rightarrow & \{1\} \neq\{1,1\}
\end{aligned}
$$

## Part c) (5 points)

Set: To be part of the result, a tuple must meet either condition C or D (or both).
Thus, if we select out tuples in $R$ that meet condition $C$ and meet condition $D, a$ result tuple must be in either one of the groups. The union operator then will take all the elements of one group and put them to the other with all the common elements occurring once. Thus, it does not matter when we use the union operator, making the overall rule holds for sets.

Bag: Counterexample


Conditions: $\mathrm{C}: \quad \mathrm{A}<5$
D: $\quad B>2$

$$
\sigma_{A<5 \text { OR } B>2}(R)=\sigma_{A<5}(R) \cup \sigma_{B>2}(R)
$$

$$
\begin{aligned}
& \rightarrow\{(3,5)\}=\{(3,5) \cup(3,5)\} \\
& \rightarrow\{(3,5)\} \neq\{(3,5),(3,5)\}
\end{aligned}
$$

Problem 5.4.1 (24 points)
Part b) (3 points)
Part d) (3 points)

| B | C |
| :---: | :---: |
| $\cdots-\cdots-----------1$ |  |
| 0 | 1 |
| 0 | 2 |
| 2 | 4 |
| 2 | 5 |
| 3 | 4 |
| 3 | 4 |

Part h) (3 points)

| B | AVG(C) |
| :---: | :---: |
| 0 | 1.5 |
| 2 | 4.5 |
| 3 | 4 |


| Part i) |  | (3 points) | Part j) |  | (3 points) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A |  |  | A |  | $\operatorname{MAX}(\mathrm{C})$ |
| ---- |  | (3 points) |  |  |  |
| 0 |  |  | 2 |  | 4 |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| Part I) |  |  | Part m) |  | (3 points) |
| A | B | C | A | B | C |
| 2 | 3 | 4 | 2 | 3 | 4 |
| 2 | 3 | 4 | 2 | 3 | 4 |
| NULL | 0 | 1 | 0 | 1 | NULL |
| NULL | 2 | 4 | 0 | 1 | NULL |
| NULL | 2 | 5 | 2 | 4 | NULL |
| NULL | 0 | 2 | 3 | 4 | NULL |
|  |  |  | NULL | 0 | 1 |
|  |  |  | NULL | 2 | 4 |
|  |  |  | NULL | 2 | 5 |
|  |  |  | NULL | 0 | 2 |

## Problem 5.4.2 (15 points)

Part c) (5 points)

## I dempotent

Selection is based on a certain condition. If a tuple meets that condition, it will be selected. Thus, regardless of how many times the operation is performed or in what order, the result will always be the same.

Part d) (5 points)

## Not idempotent

Consider the following counterexample:
A B

1) $\gamma_{A, \operatorname{CoUNT}(A)} \rightarrow(1,3)$
11
12
2) $\gamma_{A, \operatorname{CouNT}(A)} \rightarrow(1,1)$

Part e) (5 points)

## I dempotent

Sorting does not change the contents of the table. Even if there were duplicates, it is presumed the algorithm used to sort duplicates is deterministic (not random). That is, it will always sort the same way. Thus, applying the same consistent algorithm over and over will yield the same result.

