Sheet 12: Priom Biswas and Md Nabid Imteaj

PROBLEM 1:

Source Code: Submitted
Output:

```
---PersonTable---
[FID=1, dob=12.11.89, name=Hoffman, zip=66133]
[FID=2, dob=04.01.93, name=Mayer, zip=67663]
[FID=3, dob=11.11.89, name=Hofmann, zip=66123]
[FID=4, dob=01.04.1993, name=Mayer, zip=67653]
[FID=5, dob=01.04.1993, name=Meyer, zip=67663]
-----------------

---ProbabilisticPersonTable---
[dobs={11.11.89=0.4, 12.11.89=0.6}, fids=[1, 3], names={Hoffman=0.6,
Hofmann=0.4}, zips={66123=0.4, 66133=0.6}]
[dobs={01.04.1993=0.7, 04.01.93=0.3}, fids=[2, 4, 5], names={Meyer=0.5,
Mayer=0.5}, zips={67663=0.8, 67653=0.2}]
-----------------

All checks passed, solution probably correct.
```

Problem 2

S

	A	B	
s_1	'a'	1	0.6
s_2			

T

	C	D	
t_1	1	'x'	0.5
t_3	1	'y'	0.8

$D_1 = \{s_1, t_1, t_3\} = 0.6 \times 0.5 \times 0.8 = 0.24$

$D_2 = \{s_1, t_1\} = 0.6 \times 0.5 \times (1 - 0.8) = 0.06$

$D_3 = \{s_1, t_3\} = 0.6 \times 0.5 \times 0.8 = 0.24$

$D_4 = \{t_1, t_3\} = 0.16$

$D_5 = \{s_1\} = 0.06$

$D_6 = \{t_1\} = 0.04$

$D_7 = \{t_3\} = 0.16$

$D_8 = \emptyset = 0.04$

$q^{pwd}(D) =$

answer	P
{'x', 'y'}	0.24
{'x'}	0.06
{'y'}	0.24
$\emptyset$	0.46

$\Sigma = 1.00$

$q^{rank}(z) \leftarrow S(x,y), T(y,z)$

$e = (s_1 \wedge t_1) \vee (s_1 \wedge t_3)$

s_1	t_1	t_3	e	
F	F	F	F	
F	F	T	F	
F	T	F	F	
F	T	T	F	
T	F	F	F	
T	F	T	T	$= 0.6 \times 0.5 \times 0.8 = 0.24$
T	T	F	T	$= 0.6 \times 0.3 \times 0.2 = 0.06$
T	T	T	T	$= 0.6 \times 0.5 \times 0.8 = 0.24$

$\therefore P(e) = 0.54$

A	B	
'e'	2	0.8

2	'y'	0.3

$D_8 = \{s_3, t_2\} = 0.24$

$D_9 = \{s_3\} = 0.56$

$D_{10} = \{t_2\} = 0.06$

$D_{11} = \emptyset = 0.14$

$q^{pwd}(D) =$

answer	P
{'y'}	0.24
$\emptyset$	0.76

$\Sigma = 1.00$

$q^{rank}(z) \leftarrow S(x,y), T(y,z)$

$e = (s_3 \wedge t_2)$

s_3	t_2	e	
F	F	F	
F	T	F	
T	F	F	
T	T	T	$= 0.8 \times 0.3 = 0.24$

$\therefore P(e) = 0.24$

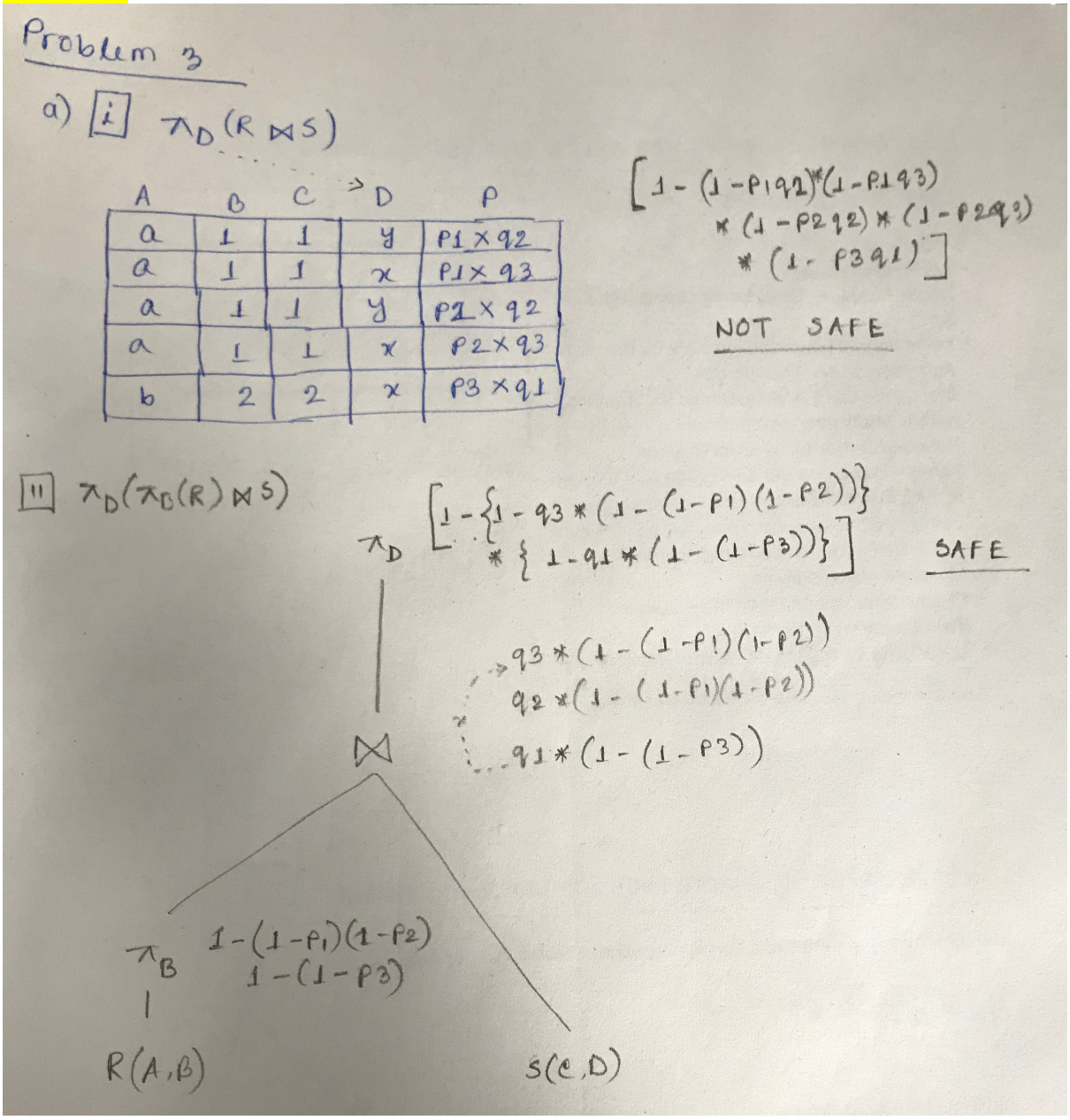
$q^{rank}(D) =$

answer	P
'x'	0.30
'y'	0.48

$q^{rank}(D) =$

answer	P
'y'	0.24

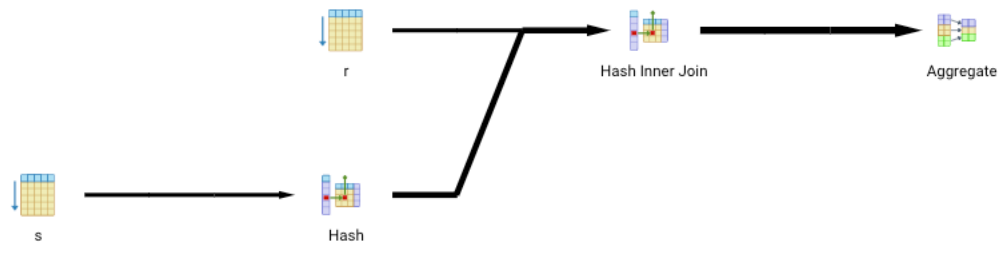
PROBLEM 3:



b)

not safe	

`select 1-prod(1-r.p*s.p) as prod`
`from r, s`
`where r.x=s.x`
	0.80368740352

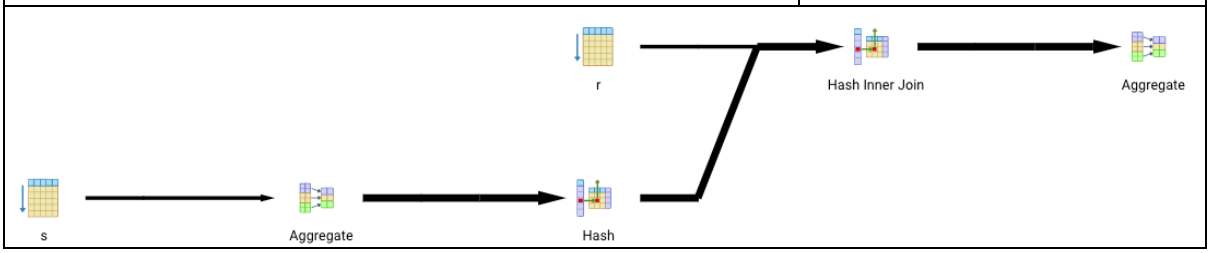
	

safe

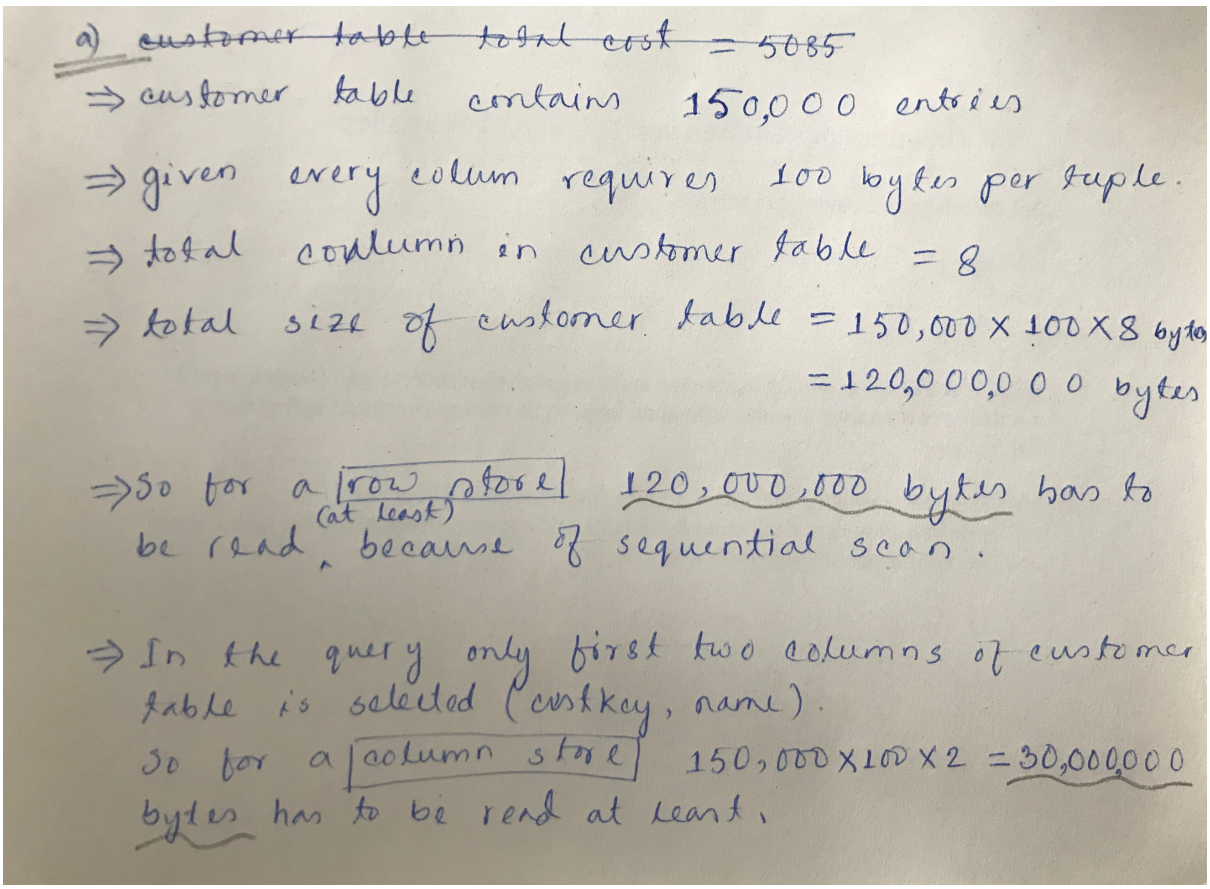
```
select ss.x, r.z, 1-(prod(1-(r.p * ss.p))) as
prod
from r, (select s.x, prod(1-s.p) as p from s
group by s.x) as ss
where r.x=ss.x
GROUP BY ss.x, r.z
```

	x charact	z charac	prod double precision
1	a4	c	0.14400000000000002
2	a2	c	0.08399999999999996
3	a1	c	0.20999999999999996
4	a3	c	0.25

Total ~= 0.686



PROBLEM 4:



a) ~~customer table total cost = 5085~~

⇒ customer table contains 150,000 entries

⇒ given every colum requires 100 bytes per tuple.

⇒ total coulumn in customer table = 8

⇒ total size of customer table = 150,000 × 100 × 8 byte
= 120,000,000 bytes

⇒ So for a row store (at least) 120,000,000 bytes has to be read, because of sequential scan.

⇒ In the query only first two columns of customer table is selected (custkey, name).
So for a column store 150,000 × 100 × 2 = 30,000,000 bytes has to be read at least.

b) ⇒ If we apply vertical partitioning for each column in customer table:

> for row store, we need 150,000 × 100 × 2 = 30,000,000 bytes to read at least, which is 75% less than previous read.

> for column store, the read will remain same due to first two column selection.

PROBLEM 5:

Problem 5

a)
```
map (string key, string value) {
    for each word w1 and next word w2 in value
        emit ({w1, w2}, 1)
}

reduce (string key, int values) {
    for each value in values
        add_edge (key.w1, key.w2)
}
```

⇒ mapper will be called for each line.

⇒ reducer will be called for each key of emitted by the mapper.

b) map (string key, string value) {

for each word w in value

emit (w, 1)

for each word w_1 and previous word w_2 in value

emit ({w_1, w_2}, 1)

}

reduce (string key, List<values>) {

~~for each value weight value w in values~~

~~add_weight (key,~~

if key contains only w {

for each value in values

add_weight (key, value)

}

else if key contains {w_1, w_2} {

for each value in values

add_edge (w_2, w_1, value)

}

}