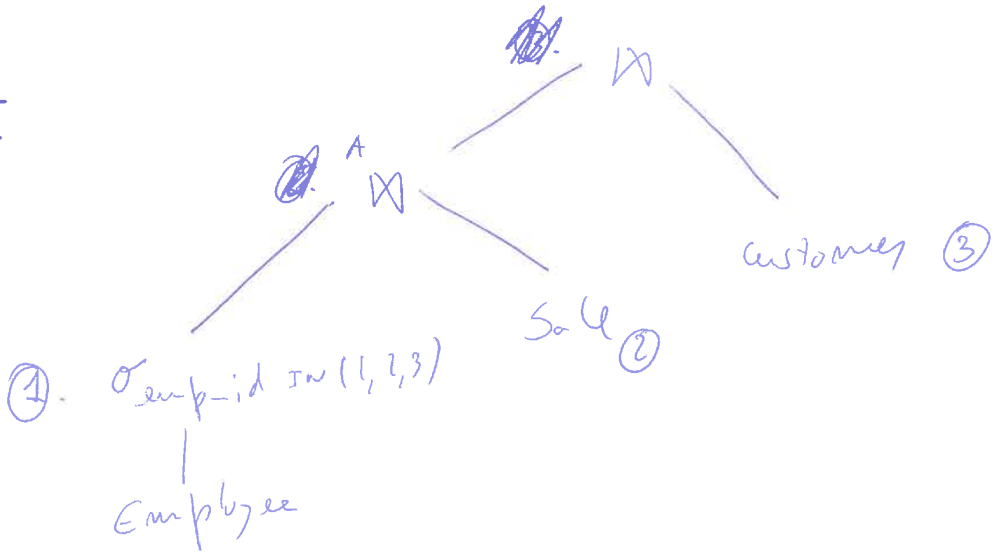
Resolução 2ª tentativa SBD 2025/26 - 15/Dez/2025

1



$M_{①} = 3$ because emp_id is PK of Employee

$M_{①⋈②} = min\left(\frac{M_{①} * M_{②}}{V(emp\_id, ①)} , \frac{M_{①} * M_{②}}{V(emp\_id, ②)} \right)$

$= Min\left(\frac{3 \times 500{,}000}{3} , \frac{3 \times 500{,}000}{100} \right)$

$= Min(500{,}000, 15000) =$

$= 15000$.

$M_{A⋈3} = 15000$ because customer_id is a foreign key in A referencing ③

$M_{A⋈③} = min\left(\frac{M_{A} * M_{Customer}}{V(customer\_id, A)} , \frac{M_{A} * M_{Customer}}{V(customer\_id, Customer)} \right)$

$= Min\left(\frac{15000 \times 20000}{V(customer\_id, A)} , \frac{15000 \times 20000}{20000} \right)$

$V(customer\_id, ①⋈②) = Min\left(V(customer\_id, Sale), M_{①⋈②} \right) = min(20.000, 15000) = 15000$

15)

	T_1	T_2
1		LOCK-X (P[486])
2	LOCK-X (P[176])	
3		LOCK-S (P[400], ..., P[499])
4	UNLOCK (P[176])	
5		LOCK-X (P[176])
6		UNLOCK (P[176], P[486], P[400], ..., P[499])

Sin bloqueo

1c)

M

D
P[127]. price = 10.50
P[128]. price = 10.50
P[129]. price = 10.70
P[130]. price = 20.50

1.
2.
3.
4. P[127]. price = 11
5. P[128] price = 12
6 P[129]. price = 10.80
7.
8. <checkpoint {T1, T3}>

P[127]. price = 11.00
P[128] price = 12.00
P[129]. price = 10.80

9. P[129]. price = 10.70
10. P[130]. price = 20.60
11. <checkpoint {T1, T3}>

P[129] price = 10.70
P[130]. price = 20.60

12.
13.

P[127] price = 11.00
P[128]. price = 12.00
P[129]. price = 10.70
P[130]. price = 20.60

14. NOTHING TO REDO IN T3
UNDO-LIST = {T1}
15. <T1, P[130]. price, 20.50>
16. <T1, P[128]. price, 10.50>
17. <T1, P[127]. price, 10.50>
18. <T1, abort>

P[130]. price = 20.50
P[128] price = 10.50
P[127]. price = 10.50

1d)

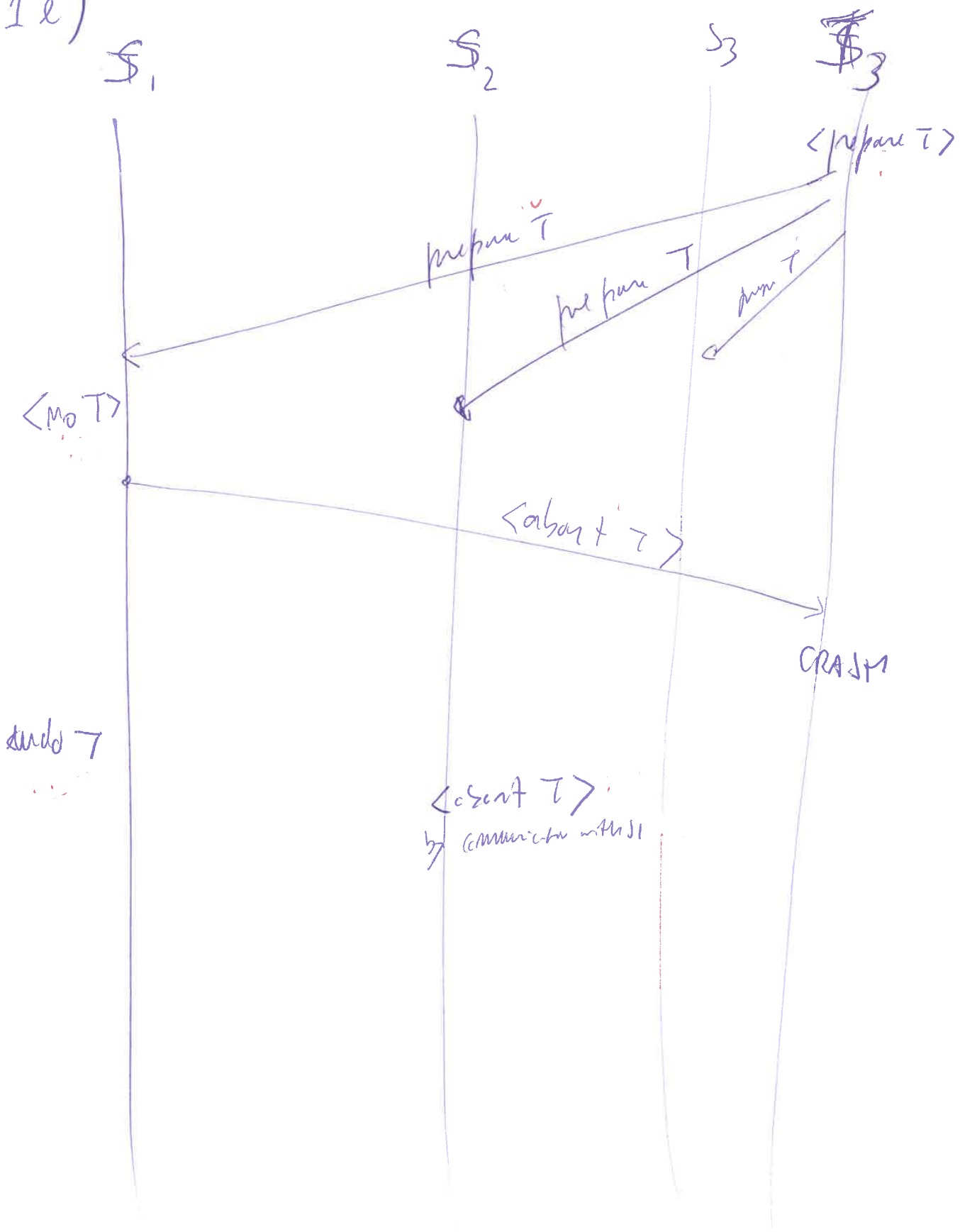
	T_1	T_2	T_3
1.		$E_2 = \{(1, 'Alan', 'administrator')\}$	
2.		$N_2 = \{1\}$	
3.	$E_1 = \{(1, 'Alice', 'admin'), (2, 'Bob', 'clerk')\}$		
4.	$N_1 = \{(2)\}$		
5.			$E_3 = \{(1, 'Alice', 'administrator'), (3, 'Charles', 'clerk')\}$
6.	$E_1 = \{(1, 'Helen', 'admin'), (2, 'Bob', 'clerk')\}$		
7.	COMMIT ✓		
8.			$N_3 = \{(2)\}$
9.			COMMIT ✓
10.		ABORT X	

$E = \{(1, 'Helen', 'admin'), (2, 'Bob', 'clerk'), (3, 'Charles', 'clerk')\}$

$N = \{(2), (2)\}$

It is not serializable since T_1, T_3 or T_3, T_1 would result in $N = \{(2), (3)\}$

1 e)


$1
$2
$3
$3
<prepare T>
prepare T
prepare T
prepare T
<NO T>
<abort T>
CRASH
decide T
<abort T>
by communication with $1

2.)

On average each value of r appears $\frac{M_S}{V(A,S)}$ times in S, so each tuple in R generates $\frac{M_S}{V(A,S)}$ results of the join, that is:

$$M_R * \frac{M_S}{V(A,S)}$$

Symetrically, each tuple of s will generate $\frac{M_R}{V(A,R)}$ tuples in the result, the total being

$$M_S * \frac{M_R}{V(A,R)}$$

The minimum is due to the fact that we want obtain among values the the least number of possible

25)

Cascading rollbacks occur when a transaction reads an item written by a transaction that it has not committed and then later aborts.

So, an Escalades scheduler if a transaction T_j reads an item written by transaction T_i, the commit operation occurs before the read of T_j

In the strict two-phase locking protocol all (exclusive) locks for reads (shared) and writes are held until the growing phase but the exclusive locks are only released after the commit.

therefore if a transaction needs to read an item written by another transaction then it can only do it after the commit of transaction since the release of the exclusive lock is done after the commit, blocking the transaction that wanted to read.

2c)

force-policy: requires updated blocks to be written at commit

so the no-force policy is less expensive at commit giving more freedom to the buffer manager to delay writes to when it is most appropriate.

steal policy: blocks containing updates of uncommitted transactions can be written to disk.

this gives the buffer manager more freedom to delay the operation of disk writing when it is most appropriate.