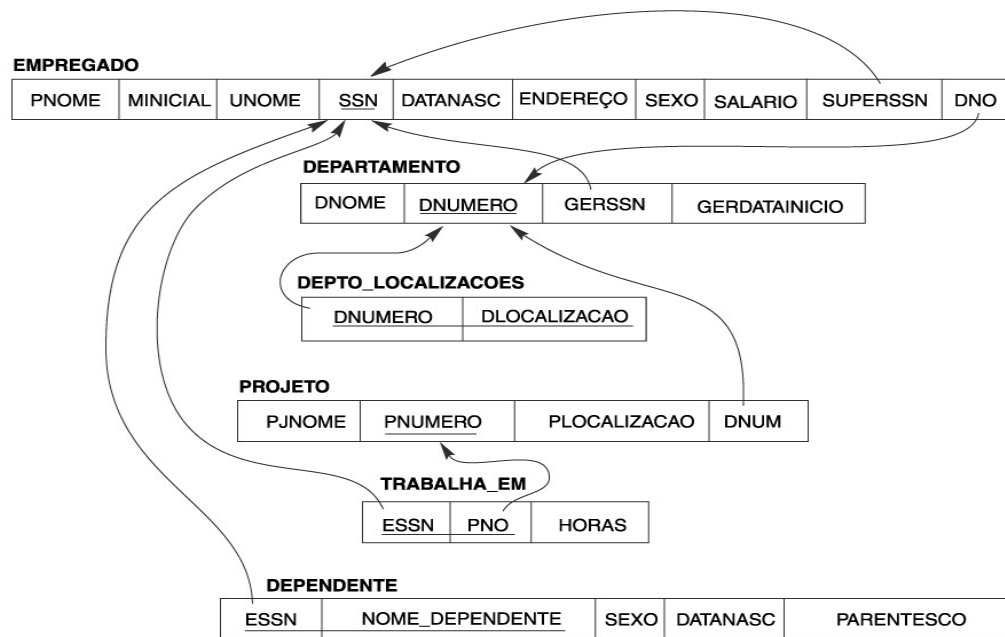


## 2ª Prova

1. (4.0 pontos) Considerando o esquema abaixo, faça a especificação das seguintes consultas em MongoDB:
  - a. Consulta para recuperar o ssn, pnome e unome dos empregados que ganham salário menor que 45000.
  - b. Consulta para recuperar os empregados e a quantidade de horas trabalhadas por cada um deles como um agregado, para os empregados do departamento 5.
  - c. Consulta para recuperar a soma da quantidade de horas trabalhadas pelos empregados em cada projeto que tenha pno maior que 5.
  - d. Consulta para recuperar os departamentos e seus respectivos empregados como agregados dos departamentos. Ordene o resultado pelo nome do departamento (campo dnome).



2. (2 pontos) Considerando as quatro transações,  $T_1$ ,  $T_2$ ,  $T_3$  e  $T_4$ , e os planos  $S_1$  e  $S_2$  abaixo, construa os grafos de precedência de para  $S_1$  e  $S_2$  e diga se são serializáveis em conflito ou não.

$T_1$ :  $r_1(X)$ ;  $r_1(Z)$ ;  $w_1(X)$ ;  $w_1(Z)$ ;  
 $T_2$ :  $r_2(Z)$ ;  $r_2(Y)$ ;  $w_2(Z)$ ;  $w_2(Y)$ ;  
 $T_3$ :  $r_3(X)$ ;  $r_3(Y)$ ;  $w_3(Y)$ ;  
 $T_4$ :  $r_4(Y)$ ;  $w_4(Y)$ ;  $r_4(Z)$ ;  $w_4(Z)$ ;

$S_1$ :  $r_1(X)$ ;  $r_2(Z)$ ;  $r_3(X)$ ;  $r_4(Y)$ ;  $w_2(Z)$ ;  $w_4(Y)$ ;  $r_2(Y)$ ;  $r_1(Z)$ ;  $w_2(Y)$ ;  $r_3(Y)$ ;  $w_1(X)$ ;  $w_1(Z)$ ;  $r_4(Z)$ ;  $w_3(Y)$ ;  $w_4(Z)$ ;  
 $S_2$ :  $r_1(X)$ ;  $r_1(Z)$ ;  $r_3(X)$ ;  $w_1(X)$ ;  $r_3(Y)$ ;  $w_1(Z)$ ;  $w_3(Y)$ ;  $r_2(Z)$ ;  $r_4(Y)$ ;  $w_4(Y)$ ;  $r_2(Y)$ ;  $w_2(Z)$ ;  $r_4(Z)$ ;  $w_2(Y)$ ;  $w_4(Z)$ ;

3. (2 pontos) A figura abaixo mostra o *log* de execução de dois escalonamentos das transações  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  e  $T_5$  até o ponto de queda do sistema. Descreva o processo para recuperação do banco utilizando os protocolos definido para cada *log*. Especifique quais operações serão refeitas ou desfeitas, e qual será o resultado final das variáveis A, B, C e D.

4.

a) Usar UNDO/REDO

```
[start_transaction, T1]
[write, T1, D, 10, 25]
[write, T1, C, 15, 60]
[start_transaction, T2]
[write, T2, B, 33, 45]
[checkpoint, T1, T2]
[start_transaction, T3]
[write, T3, B, 45, 52]
[commit, T1]
[write, T2, A, 11, 27]
[commit, T2]
[write, T3, C, 60, 42]
[commit, T3]
[start_transaction, T4]
[write, T4, A, 27, 50]
[start_transaction, T5]
[write, T4, D, 25, 75]
[write, T5, C, 42, 53]
[write, T5, B, 52, 78]
[write, T4, C, 53, 82] ← Queda do Sistema
```

a) Usar NO-UNDO/REDO

```
[start_transaction, T1]
[write, T1, A, 5, 30]
[write, T1, C, 2, 28]
[start_transaction, T2]
[write, T1, D, 20, 45]
[commit, T1]
[write, T2, B, 12, 38]
[write, T2, A, 30, 42]
[start_transaction, T3]
[checkpoint, T2, T3]
[write, T3, D, 45, 38]
[commit, T2]
[write, T3, C, 28, 64]
[start_transaction, T4]
[write, T4, A, 42, 50]
[write, T3, D, 38, 53]
[write, T4, B, 38, 78]
[commit, T3]
[write, T4, A, 50, 85]
[commit, T4]
[start_transaction, T5] ← Queda do Sistema
```

5. (2 pontos) Com base nas consultas abaixo faça a representação da árvore algébrica otimizada utilizando as regras de equivalência da álgebra relacional.

Consultas:

- Select e.pnome, e.unome, s.pnome, s.unome, d.nome\_dependente  
from empregado e, empregado s, dependente d  
where e.superssn = s.ssn and d.essn = e.ssn and e.salario > ( Select AVG(salario) from empregado  
where superssn is not null)
- Select d.dnome, p.pjnome, e.pnome, e.unome, SUM(t.horas)  
from departamento d join projeto p on (d.dnumero = p.dnum) join trabalha\_em t on (p.pnumero =  
t.pno) join empregado e on (e.dno = d.dnumero)  
where d.dnome = 'Pesquisa' and t.horas > 15  
group by d.dnome, p.pjnome, e.pnome, e.unome