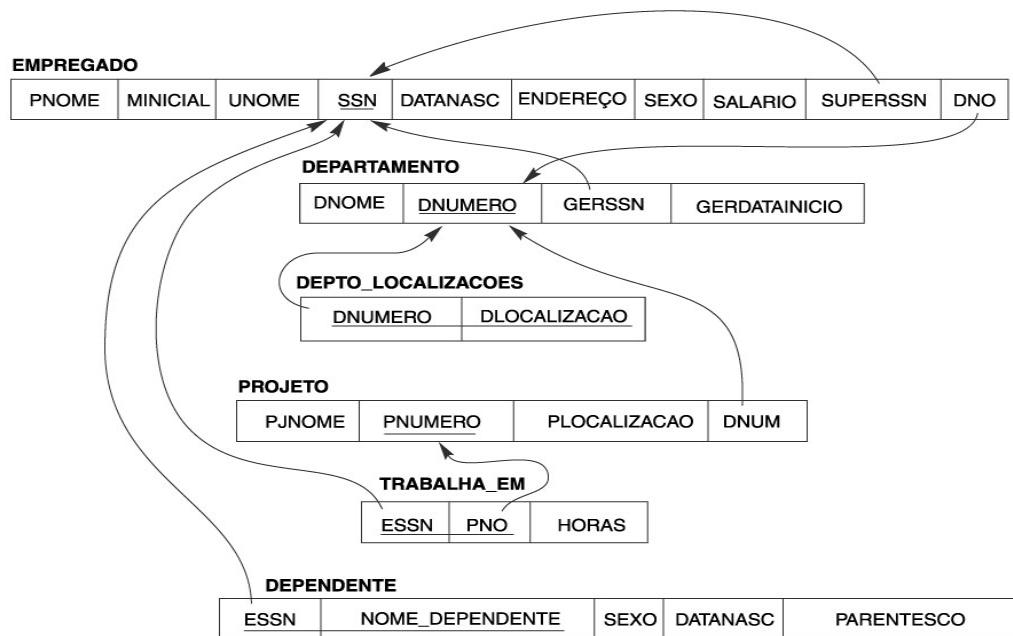


2ª Prova

1. (6.0 pontos) Considerando o esquema abaixo, faça a especificação das seguintes consultas em MongoDB:

- Consulta que recupere o nome, o ssn e o salario dos empregados que ganham mais de 30000 ordenados por salário em ordem ascendente.
- Consulta que recupere o nome, ssn e o sexo dos empregados que trabalham no departamento de Administração.
- Consulta que retorne o nome do empregado, nome do projeto e a quantidade de horas por ele trabalhada em cada projeto.
- Consulta que retorne a quantidade e a média do salário dos empregados por departamento e mostre no resultado o nome do departamento, a quantidade e a média.
- Consulta que recupere a soma da quantidade de horas trabalhadas por cada empregado em projetos e mostre o nome do projeto e a quantidade de horas.
- Consulta que recupere a quantidade de horas e a quantidade de empregado dos projetos de cada departamento e mostre o nome do departamento, a quantidade de empregado e a quantidade de horas.



2. (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.

a) Usar UNDO/REDO

```
[start_transaction, T1]
[write, T1, D, 10, 34]
[write, T1, C, 15, 53]
[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, 53, 42]
[commit, T3]
[start_transaction, T4]
[write, T4, A, 27, 50]
[start_transaction, T5]
[write, T4, D, 34, 75]
[write, T5, C, 42, 53]
[start_transaction, T6]
[write, T5, B, 52, 78]
[write, T4, C, 53, 82]
[write, T6, A, 50, 65]
[commit, T4]
[write, T5, B, 78, 96]
[write, T6, D, 75, 26]
[write, T6, C, 82, 15]
```

Queda do Sistema
←—————

a) Usar UNDO/NO-REDO

```
[start_transaction, T1]
[write, T1, A, 5, 34]
[write, T1, C, 2, 28]
[start_transaction, T2]
[write, T1, D, 20, 40]
[commit, T1]
[write, T2, B, 12, 44]
[write, T2, A, 34, 42]
[start_transaction, T3]
[checkpoint, T2, T3]
[write, T3, D, 40, 44]
[commit, T2]
[write, T3, C, 28, 64]
[start_transaction, T4]
[write, T4, A, 42, 33]
[write, T3, D, 44, 53]
[start_transaction, T5]
[write, T4, B, 44, 90]
[write, T5, C, 64, 34]
[start_transaction, T6]
[write, T6, B, 90, 54]
[commit, T3]
[write, T4, A, 33, 70]
[write, T5, D, 53, 67]
[write, T6, C, 34, 24]
[write, T4, B, 54, 98]
```

Queda do Sistema
←—————

3. (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, departamento depart
where e.superssn = s.ssn and d.essn = e.ssn and e.dno = depart.dnumero and e.salario > 10000
- 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 > 10
group by d.dnome, p.pjnome, e.pnome, e.unome