

Usando o esquema abaixo, crie expressões algébricas para as questões a seguir.

Cidade (codCid, nome, uf)

Funcionario (codFun, nome, idade, fone, CPF, dataAdm, codCid)

Codcid referencia cidade (codCid)

Carro (codCarro, modelo, marca, cor, codCidPlaca)

CodcidPlaca referencia cidade (codCid)

Cliente (codcli, nome, idade, codCid, codpaicli, CPF)

Codpaicli referencia cliente (codcli)

Codcid referencia cidade (codCid)

Reserva (codcar, codcli, codFun, data)

codFun referencia funcionario (codFun)

codCar referencia carro (codCarro)

codCli referencia cliente (codCli)

- Seleção -  $\sigma$
- Projeção -  $\pi$
- Produto Cartesiano - X
- Renomeação -  $\rho$
- Junção -  $|X|$
- Diferença - -
- União - U
- Intersecção
- Divisão

**Construir a expressão algébrica, use operador de junção:**

1. Marca dos carros com reserva em '10/10/2000'.

```
 $\pi$  marca
(
   $\sigma$  reserva.data = '01/01/2000'
  (carro  $|X|$  codcar = codcarro reserva)
)
```

2. Código dos clientes que possuem reserva em 01/01/2000.

```
 $\pi$  codcli
(
   $\sigma$  reserva.data = '01/01/2000'
  (reserva)
)
```

3. Data da reserva e o nome do cliente que reservou o carro 'Marea'.

a. Usando JOIN com a condição de junção

```
 $\pi$  reserva.data, cliente.nome
(
   $\sigma$  carro.marca = 'Marea'
  ((carro  $|X|$  codcar = codcarro reserva)  $|X|$  reserve.codcli = cliente.codcli cliente)
)
```

- b. Usando NATURAL JOIN (se possível)

```

π reserva.data, cliente.nome
(σ carro.marca = 'Marea'
  ((carro |X| codcar = codcarro reserva) |X| cliente)
)

```

- c. Usando PRODUTO CARTESIANO

```

π reserva.data, cliente.nome
(σ carro.marca = 'Marea' ^ codcar = codcarro ^ cliente.codcli = reserva.codcli
  ((carro X reserva) X cliente)
)

```

4. Nome e idade dos clientes e funcionários que têm menos de 30 anos (usar operador de conjunto)

```

π nome, idade
σ idade < 30 (cliente)
U
π nome, idade
σ idade < 30 (funcionario)

```

5. Nomes das cidades cujos moradores (os clientes) não efetuaram reserva no ano de 2003.

```

π cidade.nome (cidade |X| cliente)
  |X|
    (π codcli (cliente)
      -
      π codcli
        σ data >= '01/01/2001' ^ data <= '31/12/2003' (reserva))

```

6. Nome e idade dos clientes, e data de suas reservas. Clientes sem reserva também devem aparecer no resultado.

```

π cidade.nome, cliente.idade, reserva.data
(cliente = |X| reserva)

```

7. Código dos clientes que possuem idade superior a 30 anos e que não efetuaram nenhuma reserva (usar operador de conjunto).

```

π cliente.codigo
(σ idade > 30
  (cliente)
  -
  π cliente.codigo
    (reserva))

```

8. Nomes dos clientes com mais de 30 anos com reservas marcadas para o dia 01/01/2000.

```
π cliente.nome  
  (σ idade > 30 ^ data = '01/01/2000'  
    (reserva) |X| cliente)  
  )
```

9. Nomes dos clientes e nomes dos seus pais. Clientes cujo pai não foi informado também devem aparecer no resultado.

```
π cli.nome, pai.nome  
  (ρ cli (cliente) = |X| cli.codpaicli = pai.codcli ρ pai (cliente)))
```

10. Modelo do carro, nome do cliente e nome do funcionário que efetuou reserva no período de 01/01/1999 até 31/12/2005.

```
π carro.modelo, cliente.nome, funcionario.nome  
  (σ reserve.data >= '01/01/1999' ^ reserve.data <= '31/12/2005'  
    (((carro |X| codcar = codcarro reserva) |X| cliente) |X| funcionario)  
  )
```

11. Nome do cliente e nome do seu pai. O cliente deve ser maior de 18 anos e o pai maior de 50 anos.

```
π cli.nome, pai.nome  
  (σ cli.idade > 18 ^ pai.idade > 50  
    (ρ cli (cliente) = |X| cli.codpaicli = pai.codcli ρ pai (cliente)))
```

12. Modelo e marca dos carros que foram reservados no ano de 2000, por clientes que moram em SC e cujo funcionário que realizou a reserva foi admitido no ano 2000 (mesmo ano da reserva).

a. Usando JOIN com a condição de junção

```
π carro.modelo, carro.marca  
  (σ reserve.data >= '01/01/2000' ^ reserve.data <= '31/12/2000' ^ cidade.uf = 'SC' ^  
    funcionario.dataAdm >= '01/01/2000' ^ funcionario.dataAdm <= '31/12/2000'  
    ((carro |X| codcar = codcarro reserva) |X| reserva.codfun = funcionario.codfun funcionario)  
    |X| cidade.codcid = funcionario.codcid cidade  
  )
```

b. Usando NATURAL JOIN (se possível)

```

π carro.modelo, carro.marca
(σ reserve.data >='01/01/2000' ^ reserve.data <='31/12/2000' ^ cidade.uf = 'SC' ^
  funcionario.dataAdm >='01/01/2000' ^ funcionario.dataAdm <='31/12/2000'
  ((carro |X| codcar = codcarro reserva) |X| funcionario) |X| cidade
)

```

c. Usando PRODUTO CARTESIANO

```

π carro.modelo, carro.marca
(σ reserve.data >='01/01/2000' ^ reserve.data <='31/12/2000' ^ cidade.uf = 'SC' ^
  funcionario.dataAdm >='01/01/2000' ^ funcionario.dataAdm <='31/12/2000' ^
  codcar = codcarro ^ reserva.codfun = funcionario.codfun ^
  cidade.codcid = funcionario.codcid
  ((carro |X| reserva) |X| funcionario) |X| cidade)

```

13. Nome e UF das cidades que não tiveram nenhuma reserva de carro no mês de maio de 2000.

A questão está mal escrita. Não está claro que é cidade da placa do carro, cidade do cliente que fez a reserva.

14. Modelo e cor dos carros reservados para clientes que moram em cidades diferentes da cidade que consta na placa do carro (reservado pelo cliente).

```

π carro.modelo, carro.cor
(σ cidade.codcid <> carro.codcidplaca
  (((carro |X| carro.codcarro = reserva.codcar reserva)
   |X| reserva.codcli = cliente.codcli cliente) X cidade
)

```

15. Nome dos funcionários que moram nas mesmas cidades dos clientes para os quais eles fizeram reservas.

```

π funcionario.nome
(σ cliente.codcid = funcionario.codcid
  (((cidade |X| cidade.codcid = funcionario.codcid funcionario)
   |X| cidade.codcid = cliente.codcid cliente)
   |X| reserva.codcli = cliente.codcli ^ funcionario.codfun = reserva.codfun reserva)
)

```