

*Exercícios de Revisão para a Prova 1***Consultas em Álgebra Relacional**

Exercício 1 - Considere as seguintes relações:

Fornecedores(codf: inteiro, nome: string, endereco: string)

Pecas(codp: inteiro, nome: string, cor: string)

Catalogo(codf: inteiro, codp: inteiro, preco: real)

codf: é uma chave estrangeira para Fornecedores

codp: é uma chave estrangeira para Pecas

Os campos-chave encontram-se sublinhados, e o domínio de cada campo encontra-se listado após o nome do campo. Assim, codf é a chave de Fornecedores, codp é a chave de Peças e a composição de codf e codp formam a chave de catálogo. A relação Catalogo lista os preços cobrados por peças pelos Fornecedores. Escreva as seguintes consultas em álgebra relacional e cálculo relacional de tuplas. Obs.: As consultas de (m) a (o) só podem ser respondidas em álgebra relacional

Resposta em Álgebra Relacional:

a) Encontre os nomes dos fornecedores que fornecem alguma peça vermelha.

$\pi_{\text{Fornecedores.nome}} (\text{Fornecedores} * (\text{Catalogo} * (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas}))))$

b) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha ou verde.

$\pi_{\text{codf}} (\text{Catalogo} * (\sigma_{\text{cor} = \text{'vermelha'}} \text{ OR } \text{cor} = \text{'verde'}} (\text{Pecas})))$

c) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha ou que estão no endereço Av. Paulista, 572.

$\pi_{\text{codf}} (\text{Catalogo} * (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas}))) \cup \pi_{\text{codf}} (\sigma_{\text{endereco} = \text{'Av. Paulista, 572'}} (\text{Fornecedores}))$

d) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha e alguma peça verde.

$\pi_{\text{codf}} (\text{Catalogo} * (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas}))) \cap \pi_{\text{codf}} (\text{Catalogo} * (\sigma_{\text{cor} = \text{'verde'}} (\text{Pecas})))$

e) Encontre os códigos dos fornecedores que fornecem todas as peças.

$\pi_{\text{codf,codp}} (\text{Catalogo}) \div \pi_{\text{codp}} (\text{Pecas})$

f) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas.

$\pi_{\text{codf,codp}} (\text{Catalogo}) \div \pi_{\text{codp}} (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas}))$

g) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas ou verdes.

$\pi_{\text{codf,codp}} (\text{Catalogo}) \div \pi_{\text{codp}} (\sigma_{\text{cor} = \text{'vermelha'}} \text{ OR } \text{cor} = \text{'verde'}} (\text{Pecas}))$

h) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas ou fornecem todas as peças verdes.

$\pi_{\text{codf,codp}} (\text{Catalogo}) \div \pi_{\text{codp}} (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas})) \cup$

$\pi_{\text{codf,codp}} (\text{Catalogo}) \div \pi_{\text{codp}} (\sigma_{\text{cor} = \text{'verde'}} (\text{Pecas}))$

- i) Encontre os pares de códigos de fornecedor em que o primeiro fornecedor do par cobre mais por alguma peça que o segundo fornecedor do par.

$$\pi_{\text{codf, codf2}} (\sigma_{\text{codp} = \text{codp2 AND preco} > \text{preco2}} (\text{Catalogo} \times (\rho_{\text{Catalogo2}(\text{codf2, codp2, preco2})} (\text{Catalogo}))))$$

- j) Encontre os códigos das peças fornecidas por pelo menos dois fornecedores diferentes.

$$\pi_{\text{codp}} (\sigma_{\text{codp} = \text{codp2 AND codf} \neq \text{codf2}} (\text{Catalogo} \times (\rho_{\text{Catalogo2}(\text{codf2, codp2, preco2})} (\text{Catalogo}))))$$

- k) Encontre os códigos das peças mais caras fornecidas pelo fornecedor chamado Bill Gates.

$$\pi_{\text{codp}} (\sigma_{\text{codf} = \text{codf_bill AND preco} = \text{maior_preco}} (\text{Catalogo} \times (\rho_{(\text{codf_bill, maior_preco})} (\text{codf } \mathcal{F}_{\text{MAX preco}} (\text{Catalogo} * (\sigma_{\text{nome} = \text{'Bill Gates'}} (\text{Fornecedores})))))))$$

- l) Encontre os códigos das peças fornecidas por todos os fornecedores por menos de R\$200,00. (Se algum fornecedor não fornece a peça ou cobra mais do R\$200,00 por ela, a peça não é selecionada).

$$(\pi_{\text{codp, codf}} (\sigma_{\text{preco} < 200} (\text{Catalogo})) \div \pi_{\text{codf}} (\text{Fornecedores}))$$

- m) Encontre o número total de peças no catálogo.

$$\mathcal{F}_{\text{CONT codp}} (\text{Catalogo})$$

- n) Para cada fornecedor, encontre a quantidade de peças vermelhas fornecidas e o preço médio dessas peças.

$$\text{codf } \mathcal{F}_{\text{CONT codp, MEDIA preco}} (\text{Catalogo} * (\sigma_{\text{cor} = \text{'vermelha'}} (\text{Pecas})))$$

- o) Liste todos os pares [nome_peça, nome_fornecedor] tais que nome_peça é o nome de uma peça fornecida pelo fornecedor de nome nome_fornecedor. O nome de uma peça deve aparecer na listagem mesmo se não houve um fornecedor que a fornece (nesse caso, o nome do fornecedor deverá aparecer como NULL).

$$\pi_{\text{nome, nomef}} (\text{Peca} \bowtie \text{Peca.codp} = \text{Forn.codp} (\rho_{\text{Forn}(\text{codf, codp, nomef})} (\pi_{\text{codf, codp, nome}} (\text{Fornecedores} * \text{Catalogo}))))$$

Resposta em Cálculo de Tuplas:

- a) Encontre os nomes dos fornecedores que fornecem alguma peça vermelha.

$$\{ \text{f.nome} \mid \text{Fornecedores}(\text{f}) \text{ AND } ((\exists \text{c})(\exists \text{p})(\text{Catalogo}(\text{c}) \text{ AND } \text{Pecas}(\text{p}) \text{ AND } \text{c.codf} = \text{f.codf} \text{ AND } \text{c.codp} = \text{p.codp} \text{ AND } \text{p.cor} = \text{'vermelho'})) \}$$

- b) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha ou verde.

$$\{ \text{c.codf} \mid \text{Catalogo}(\text{c}) \text{ AND } ((\exists \text{p})(\text{Pecas}(\text{p}) \text{ AND } \text{c.codp} = \text{p.codp} \text{ AND } (\text{p.cor} = \text{'vermelho'} \text{ OR } \text{p.cor} = \text{'verde'}))) \}$$

- c) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha ou que estão no endereço Av. Paulista, 572.

{ f.nome | Fornecedores(f) AND (f.endereco = 'Av. Paulista, 572' OR ((\exists c)(\exists p)(Catalogo(c) AND Pecas(p) AND c.codf = f.codf AND c.codp = p.codp AND p.cor = 'vermelho'))) }

- d) Encontre os códigos dos fornecedores que fornecem alguma peça vermelha e alguma peça verde.

{ c1.codf | Catalogo(c1) AND ((\exists p1)(Pecas(p1) AND c1.codp = p1.codp AND p1.cor = 'vermelho')) AND ((\exists c2)(\exists p2)(Catalogo(c2) AND Pecas(p2) AND c2.codf = c1.codf AND c2.codp = p2.codp AND p2.cor = 'verde'))) }

- e) Encontre os códigos dos fornecedores que fornecem todas as peças.

{ f.codf | Fornecedores(f) AND ((\forall p)(NOT Pecas(p) OR ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

ou

{ f.codf | Fornecedores(f) AND (NOT(\exists p)(Pecas(p) AND NOT ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

- f) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas.

{ f.codf | Fornecedores(f) AND ((\forall p)(NOT Pecas(p) OR NOT (p.cor = 'vermelha') OR ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

ou

{ f.codf | Fornecedores(f) AND (NOT(\exists p)(Pecas(p) AND p.cor = 'vermelha' AND NOT ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

- g) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas ou verdes.

{ f.codf | Fornecedores(f) AND ((\forall p)(NOT Pecas(p) OR NOT (p.cor = 'vermelha' OR p.cor = 'verde') OR ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

ou

{ f.codf | Fornecedores(f) AND (NOT(\exists p)(Pecas(p) AND (p.cor = 'vermelha' OR p.cor = 'verde') AND NOT ((\exists c)(Catalogo(c) AND c.codf = f.codf AND c.codp = p.codp)))) }

- h) Encontre os códigos dos fornecedores que fornecem todas as peças vermelhas ou fornecem todas as peças verdes.

{ f.codf | Fornecedores(f) AND (
(((∀ p1)(NOT Peças(p1) OR NOT (p1.cor = 'vermelha') OR
(((∃ c1)(Catalogo(c1) AND c1.codf = f.codf AND c1.codp = p1.codp))))
OR
(((∀ p2)(NOT Peças(p2) OR NOT (p2.cor = 'verde') OR
(((∃ c2)(Catalogo(c2) AND c2.codf = f.codf AND c2.codp = p2.codp))))
)}

ou

{ f.codf | Fornecedores(f) AND (
(NOT((∃ p1)(Peças(p1) AND p1.cor = 'vermelha' AND
NOT (((∃ c1)(Catalogo(c1) AND c1.codf = f.codf AND c1.codp = p1.codp))))
OR
(NOT((∃ p2)(Peças(p2) AND p2.cor = 'verde' AND
NOT (((∃ c2)(Catalogo(c2) AND c2.codf = f.codf AND c2.codp = p2.codp))))
)}

- i) Encontre os pares de códigos de fornecedor em que o primeiro fornecedor do par cobre mais por alguma peça que o segundo fornecedor do par.

{ c1.codf, c2.codf | Catalogo(c1) AND Catalogo(c2) AND c1.codf ≠ c2.codf AND
c1.codp = c2.codp AND c1.preco > c2.preco }

- j) Encontre os códigos das peças fornecidas por pelo menos dois fornecedores diferentes.

{ c1.codp | Catalogo(c1) AND (((∃ c2)(Catalogo(c2) AND c1.codf ≠ c2.codf AND
c1.codp = c2.codp)))}

- k) Encontre os códigos das peças mais caras fornecidas pelo fornecedor chamado Bill Gates.

{ c1.codp | Catalogo(c1) AND
(((∃ f1)(Fornecedores(f1) AND c1.codf = f1.codf AND f1.nome = 'Bill Gates'))
AND NOT((∃ c2) ((∃ f2)(Catalogo(c2) AND Fornecedores(f2) AND
c2.codf = f2.codf AND f2.nome = 'Bill Gates' AND c2.preco > c1.preco))) }

- l) Encontre os códigos das peças fornecidas por todos os fornecedores por menos de R\$200,00 . (Se algum fornecedor não fornece a peça ou cobra mais do R\$200,00 por ela, a peça não é selecionada).

{ p.codp | Peça(p) AND (((∀ f)(NOT Fornecedores(f) OR (((∃ c)(Catalogo(c) AND
c.codf = f.codf AND c.codp = p.codp AND c.preco ≤ 200)))) }

ou

{ p.codp | Peça(p) AND (NOT((∃ f)(Fornecedores(f) AND NOT (((∃ c)(Catalogo(c) AND
c.codf = f.codf AND c.codp = p.codp AND c.preco ≤ 200)))) }

m) Encontre o número total de peças no catálogo.

Não pode ser resolvido no cálculo de tuplas.

n) Para cada fornecedor, encontre a quantidade de peças vermelhas fornecidas e o preço médio dessas peças.

Não pode ser resolvido no cálculo de tuplas.

o) Liste todos os pares [nome_peça, nome_fornecedor] tais que nome_peça é o nome de uma peça fornecida pelo fornecedor de nome nome_fornecedor. O nome de uma peça deve aparecer na listagem mesmo se não houve um fornecedor que a fornece (nesse caso, o nome do fornecedor deverá aparecer como NULL).

Não pode ser resolvido no cálculo de tuplas.

Projeto Conceitual de BD

Exercício 2 - Usando o modelo Entidade-Relacionamento Estendido, construa um diagrama de esquema conceitual para o BD de uma empresa locadora de veículos.

A empresa aluga automóveis e camionetas de carga. Um veículo de locação é identificado por sua placa, mas possui outras informações importantes como número do chassi, cor, modelo e marca. Além disso, todo veículo é classificado de acordo com uma tabela de tipos da locadora. Por exemplo, o tipo A3 corresponde a automóveis pequenos, de quatro portas e com ar-condicionado, enquanto o tipo C4 corresponde a camionetas com capacidade de carga de até 1 tonelada.

Um tipo de automóvel define o tamanho (pequeno, médio ou grande), o número de portas e os acessórios disponíveis, que podem ser: ar-condicionado, rádio, GPS e câmbio automático. Já um tipo de camioneta de carga define a capacidade de carga da camioneta e a dimensão (largura e altura) da sua carroceria.

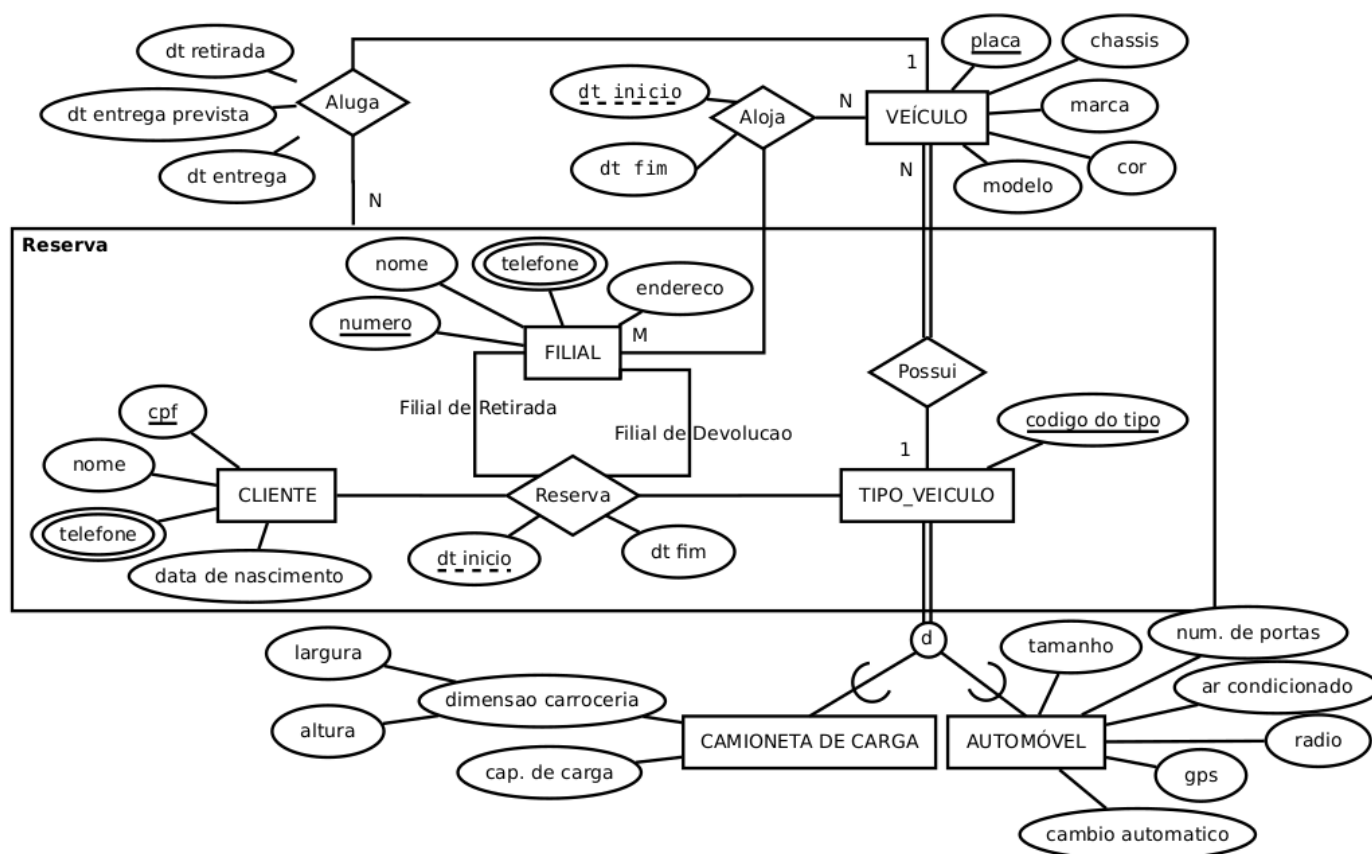
Para alugar um veículo, um cliente primeiro faz uma reserva (por telefone ou no *site* da locadora) onde define as datas de início e término da locação, o tipo de veículo que deseja alugar, a filial de retirada e a filial de devolução do veículo. E somente na retirada do veículo o cliente é “associado” a um veículo real, disponível na filial de retirada. A locadora tem uma grande rede de filiais.

De suas filiais, a locadora registra nome, número de identificação, endereço e telefones de contato. E de seus clientes, a locadora registra nome, data de nascimento, telefones e CPF.

A partir do banco de dados, devemos poder obter as seguintes informações: (i) a lista dos veículos disponíveis em cada filial na data corrente; (ii) a lista de reservas de locação ainda não atendidas em cada filial; (iii) os veículos presentemente alugados por cada filial, as suas respectivas filiais de entrega (caso sejam diferentes das de locação) e datas de entrega previstas; (iv) os cliente, as datas e locais de retirada e entrega de todas as locações já feitas na locadora.

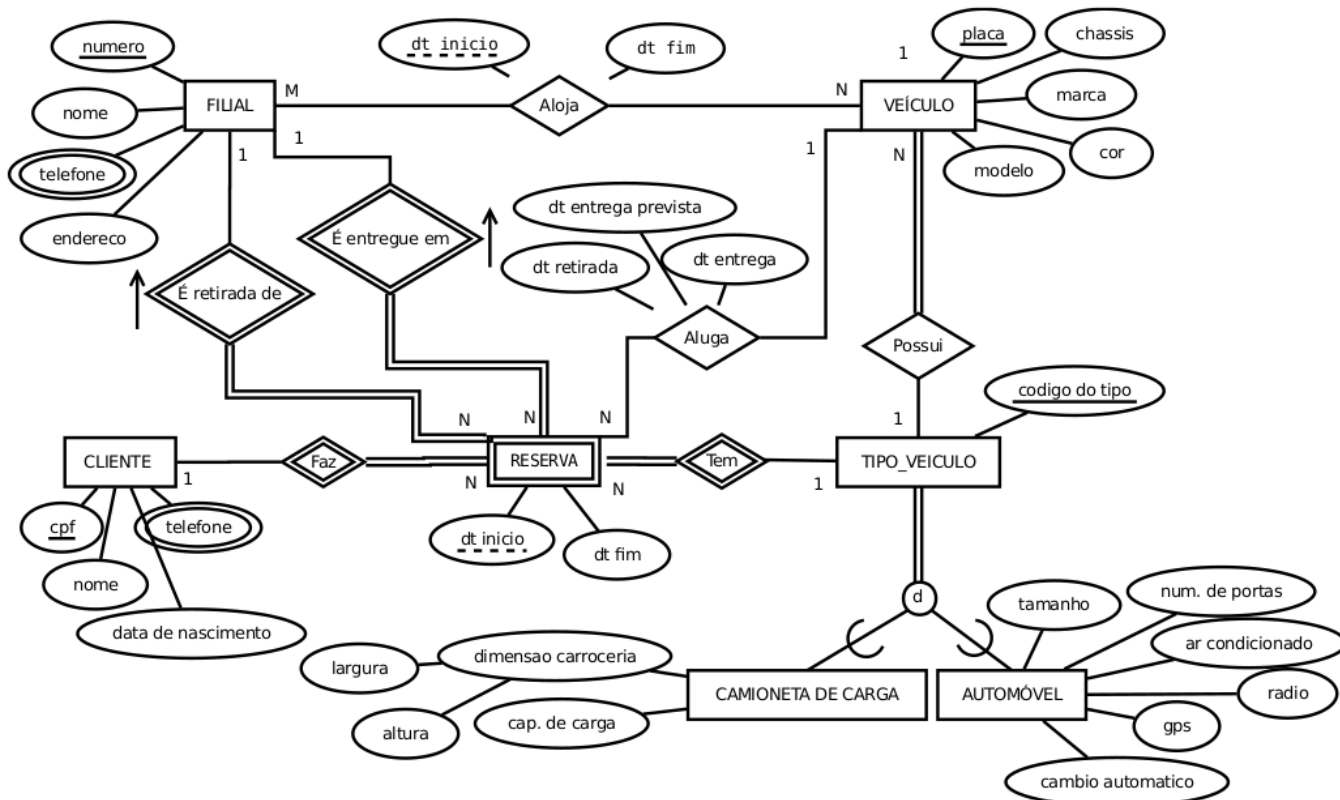
Não se esqueça de incluir no seu modelo as indicações das chaves para as entidades e das restrições de participação e de cardinalidade para os relacionamentos.

Uma possível resposta:



Obs.: Não foi indicada cardinalidade para o tipo de relacionamento Reserva (entre CLIENTE, TIPO_VEICULO, FILIAL E FILIAL) porque ele não é binário.

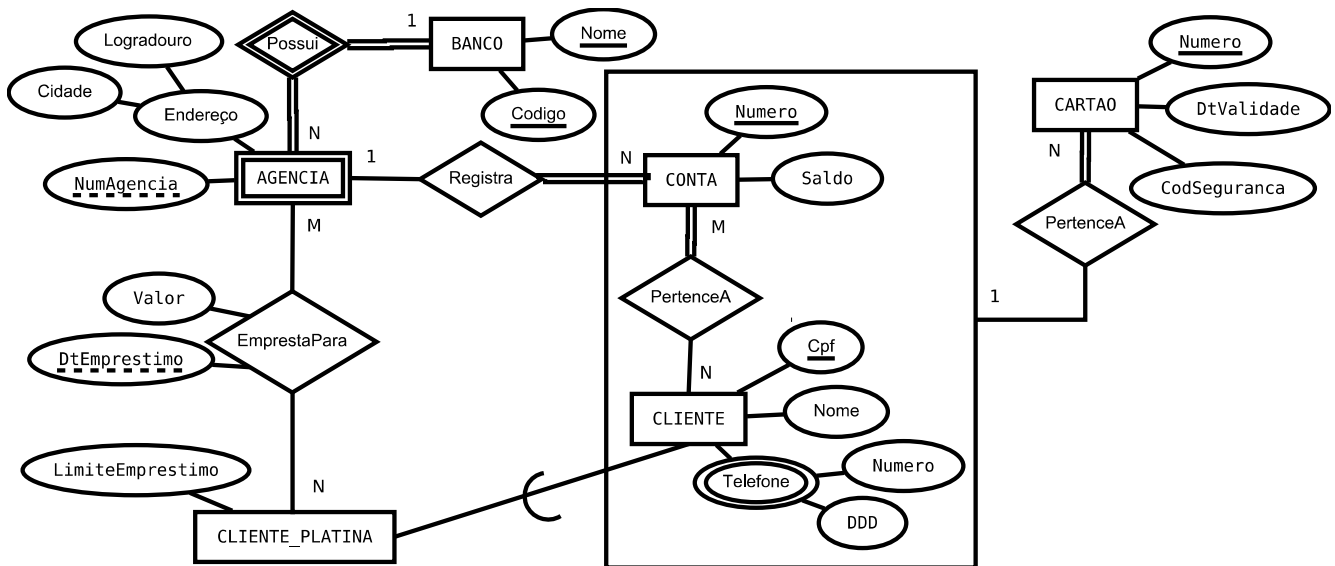
Outra possível resposta:



Obs.: Não foi indicada cardinalidade para o tipo de relacionamento Reserva (entre CLIENTE, TIPO_VEICULO, FILIAL E FILIAL) porque ele não é binário.

Projeto Lógico de BD

Exercício 3 - Crie um esquema de banco de dados relacional para o esquema EER a seguir. Não se esqueça de indicar no seu esquema as chaves primárias, chaves estrangeiras e suas respectivas ações de propagação.



Resposta:

Notação simplificada para as ações de propagação das chaves estrangeiras:

- RR: na Remoção, Restrinja (*restrict*)
- RC: na Remoção, Cascade (*cascade*)
- AR: na Alteração, Restrinja (*restrict*)
- AC: na Alteração, Cascade (*cascade*)

Mapeamento:

Nome (codBanco, nome)

---> nome é uma chave única

Agencia (codBanco, numAgencia, logradouro, cidade)

---> codBanco é uma chave estrangeira para Banco(codigo); RC

Cliente (cpf, nome)

Telefone (cpfCliente, ddd, numero)

---> cpfCliente é uma chave estrangeira para Cliente(cpf); RC

ClientePlatina (cpfCliente, limiteEmprestimo)

---> cpfCliente é uma chave estrangeira para Cliente(cpf); RC

Conta (numero, saldo, codBanco, numAgencia)

---> (codBanco, numAgencia) é uma chave estrangeira para Agencia(codBanco, numAgencia); RR

PertenceA (numConta, cpfCliente)

---> numConta é uma chave estrangeira para Conta(numero); RC

---> cpfCliente é uma chave estrangeira para Cliente(cpf); RR

EmprestaPara(codBanco, numAgencia, cpfClientePlatina, dtEmprestimo,
valor)

---> (codBanco, numAgencia) é uma chave estrangeira para
Agencia(codBanco, numAgencia); RR

---> cpfClientePlatina é uma chave estrangeira para
ClientePlatina(cpfCliente); RR

Cartao(numero, dtValidade, codSeguranca, numConta, cpfCliente)

---> (numConta, cpfCliente) é uma chave estrangeira para
PertenceA(numConta, cpfCliente); RC

SQL – Definição de Esquemas

Exercício 4 – Considere o seguinte esquema relacional mostrado a seguir. Observe que uma pessoa pode trabalhar em mais de uma empresa, tendo um salário específico para cada empresa em que trabalha.

Pessoa(pID, pNome, rua, pCidade)

Empresa(eID, eNome, eCidade)

Trabalha(pID, eID, dataInicio, salario)

Escreva comandos SQL para criar as relações acima, incluindo todas as restrições necessárias. Em particular, defina as relações de modo a garantir que:

- Toda pessoa e toda empresa tenham um nome;
- Não existam duas empresas com um mesmo nome;
- Um salário não seja inferior ao mínimo (R\$ 880,00);
- Os atributos pID e eID da relação Trabalha sejam chaves estrangeiras para, respectivamente, o atributo pID de Pessoa e o atributo eID de Empresa;
- Na remoção de uma pessoa do BD, os registros de trabalho dela sejam removidos também.

Resposta:

```
create table Pessoa(  
    pID          int primary key,  
    pNome        varchar(30) not null,  
    rua           varchar(50),  
    pCidade      varchar(30)  
);
```

```
create table Empresa(  
    eID          int primary key,  
    eNome        varchar(30) not null unique,  
    eCidade      varchar(30)  
);
```

```
create table Trabalha(  
    pID          int foreign key Pessoa(pID) on delete cascade,  
    eID          int foreign key Empresa(eID),  
    dataInicio   date,  
    salario      numeric(8,2) check (salario >= 880),  
    primary key (pID,eID)  
);
```