

Banco de Dados I

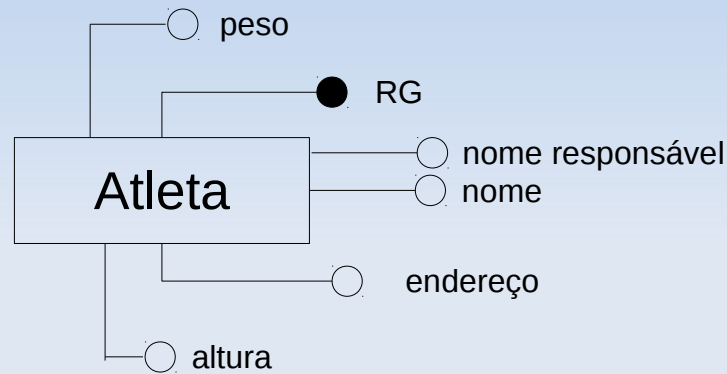
Transformação
Conceitual x Lógico Relacional

Introdução

- Etapa onde o projetista escolhe o modelo de dados do SGBD alvo (no nosso caso relacional):
 - Evitar a criação de tabelas desnecessárias: menos junções
 - Evitar atributos opcionais: desperdício de espaço nas tabelas
 - Evitar controle de restrições no BD: restringir-se às chaves e à obrigatoriedade dos atributos

Conversão

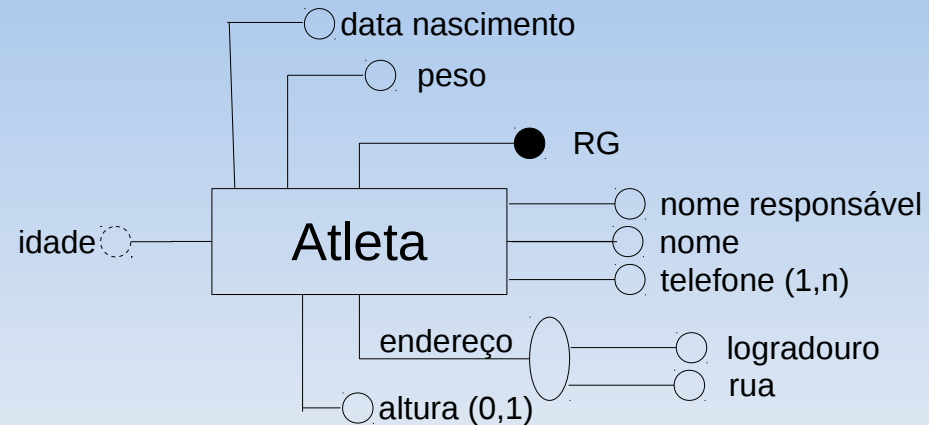
- Entidade: quase sempre é transformada em uma tabela



atleta (nrq, nome, peso, alt, nomresp, ender)

Conversão

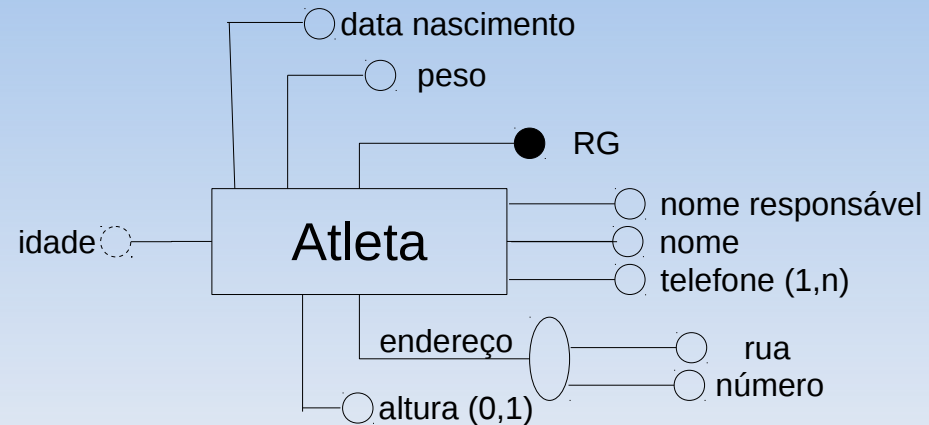
- Atributos



- Os atributos atômicos, monovalorados, opcionais e obrigatórios são criados diretamente (não esquecer de colocar o tipo)
- Para outros, tem várias opções:
- Multivalorados: nova tabela ou criação de atributos para alguns valores
- Compostos: nova tabela ou desmembramento da composição

Conversão

■ Atributos



1: atleta (nrg, nome, peso, nresp, dtnasc, rua, nro, alt, tele1, tele2, tele3)

2: atleta (nrg, nome, peso, alt, nresp, dtnasc, altura)

atleta_ender(nrg(atleta), rua, nro)

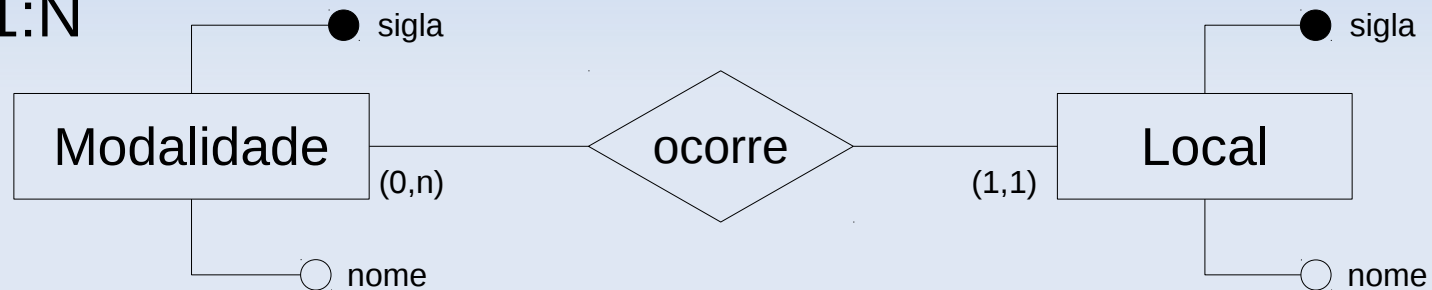
atleta_tel(nrg(atleta), nro) **ou** atleta_tel(nrg(atleta), nro)

Conversão

- Relacionamento

- Cardinalidade

- 1:N



- A entidade do lado N recebe a chave primária da entidade relacionada como chave estrangeira (lado 1 obrigatório – lado N tanto faz)

`local(sigla, nome)`

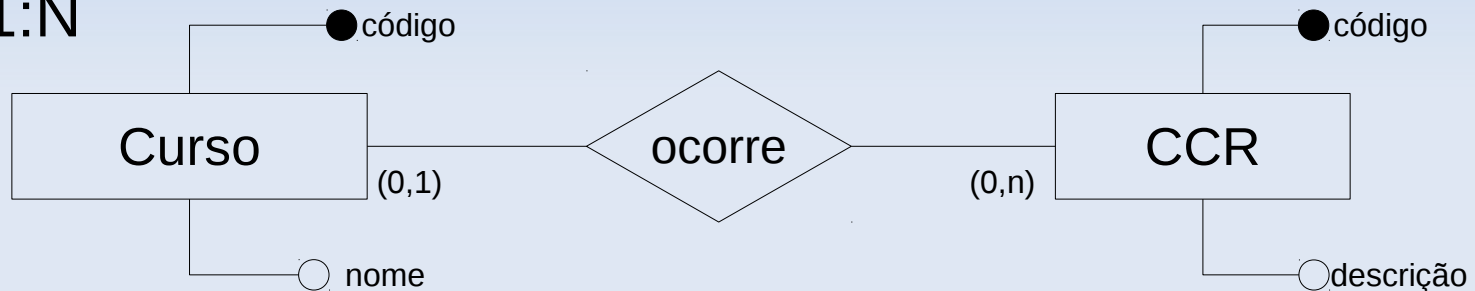
`modal(sigla, nome, sigl(local))`

Conversão

- Relacionamento

- Cardinalidade

- 1:N



- A entidade do lado N recebe a chave primária da entidade relacionada como chave estrangeira opcional (lado 1 opcional)

curso(cod, nome)

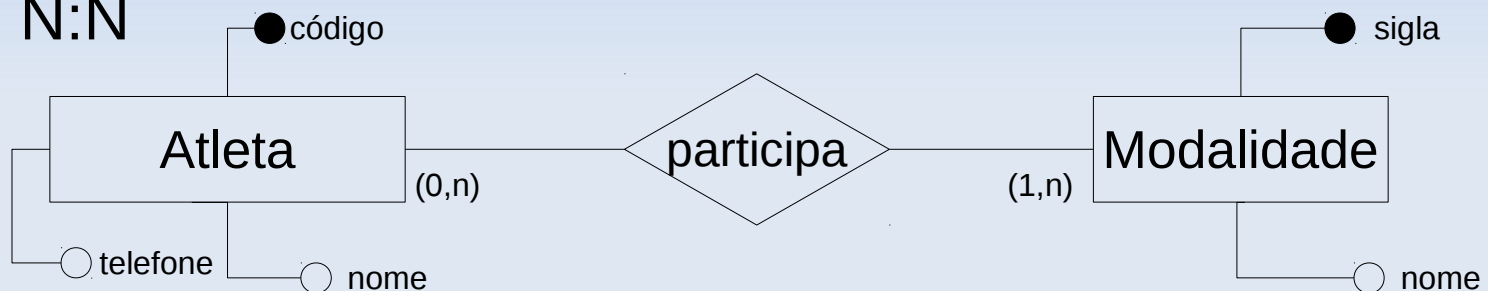
ccr(cod, descr, codc(curso))

Conversão

- Relacionamento

- Cardinalidade

- N:N



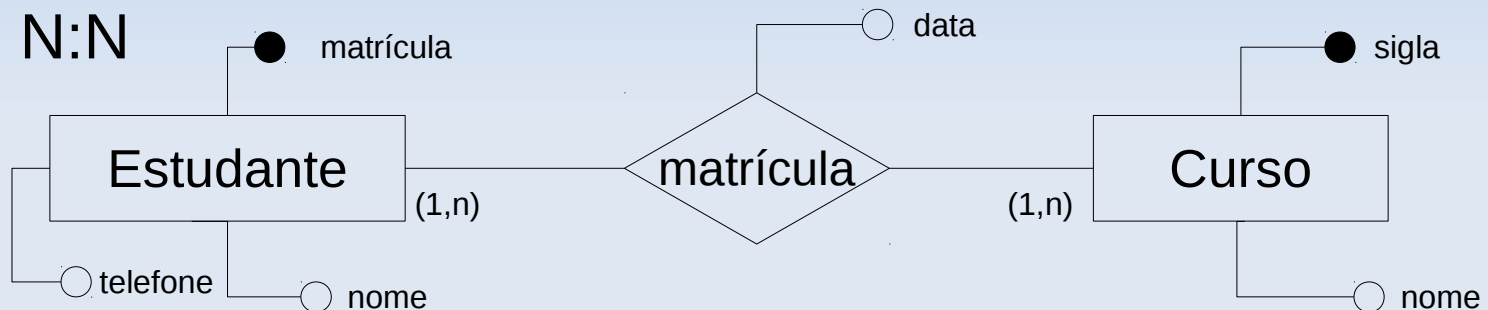
- O relacionamento vira uma tabela com as chaves das entidades envolvidas (independente da obrigatoriedade)
atleta (cod, nom, tele) modal(sigla, nome)
 atle_modal (coda(atleta), sigm(modal))
 - Os atributos transferidos se transformam em chave primária

Conversão

- Relacionamento

- Cardinalidade

- N:N



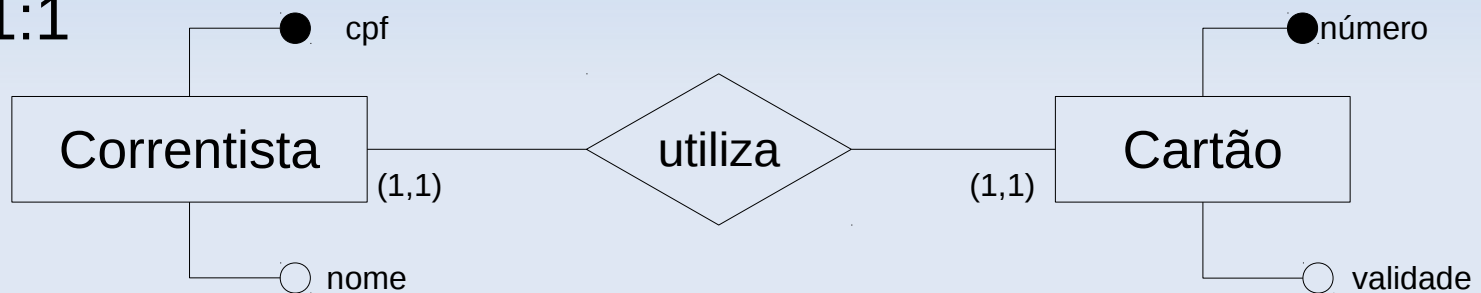
- O relacionamento vira uma tabela com as chaves das entidades envolvidas (independente da obrigatoriedade)
estudante (mat, nom, tele) curso(sigla, nome)
matricula (codc(curso), mat(estudante), data)

Conversão

- Relacionamento

- Cardinalidade

- 1:1



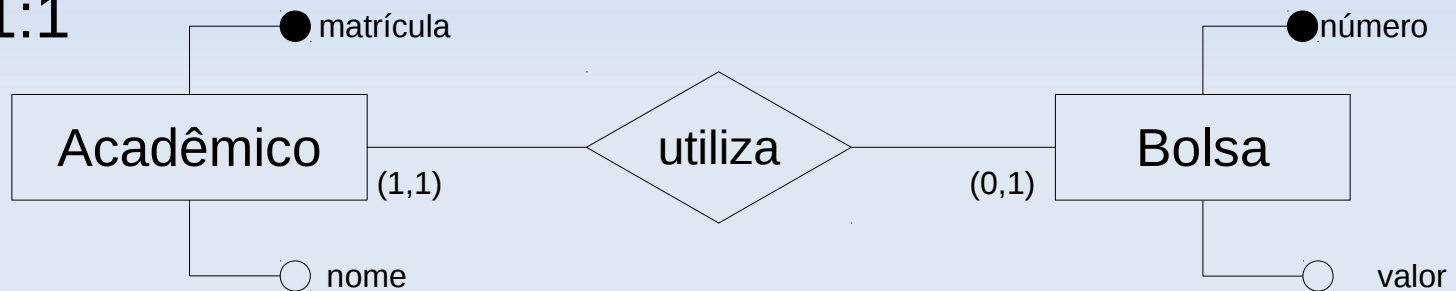
- Fusiona: `corren(cpf, nome, nrocart, valcart)`
 - Cria: `corren(cpf, nome)`
`cartao(cpf(corren), nrocart, valcart)`

Conversão

- Relacionamento

- Cardinalidade

- 1:1

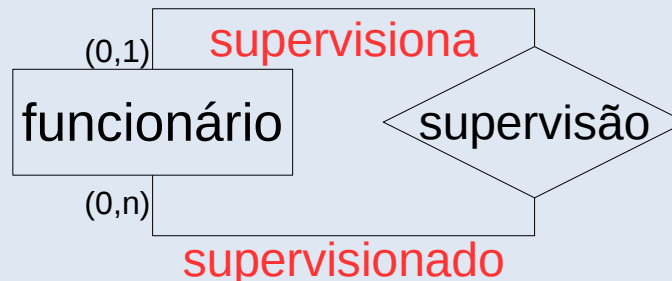


- Fusiona: `aluno(mat, nome, nbolsa, valbol)`

- Cria: `aluno(mat, nome)`
`bolsa(mat(aluno), nbolsa, valbol)`

Conversão

- Auto-relacionamentos
 - Segue a mesma lógica para relacionais entre duas entidades
 - 1:N: a entidade recebe a chave dela mesma como estrangeira
 - `func(mat, nome, dtadm, mat sup(func))`

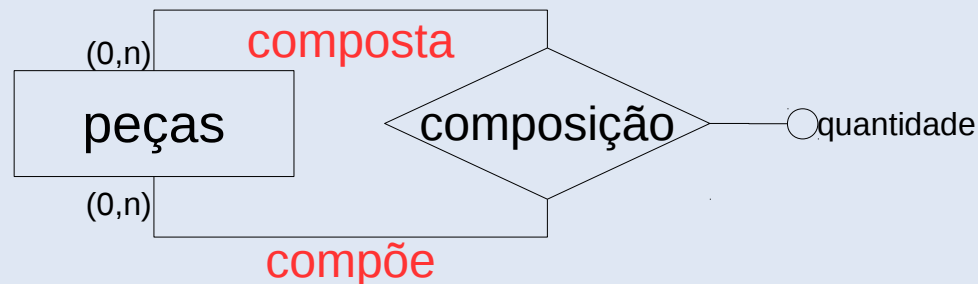


Conversão

- Auto-relacionamentos
 - Segue a mesma lógica para relacionais entre duas entidades
 - N:N: cria-se uma entidade para resolver o relacionamento

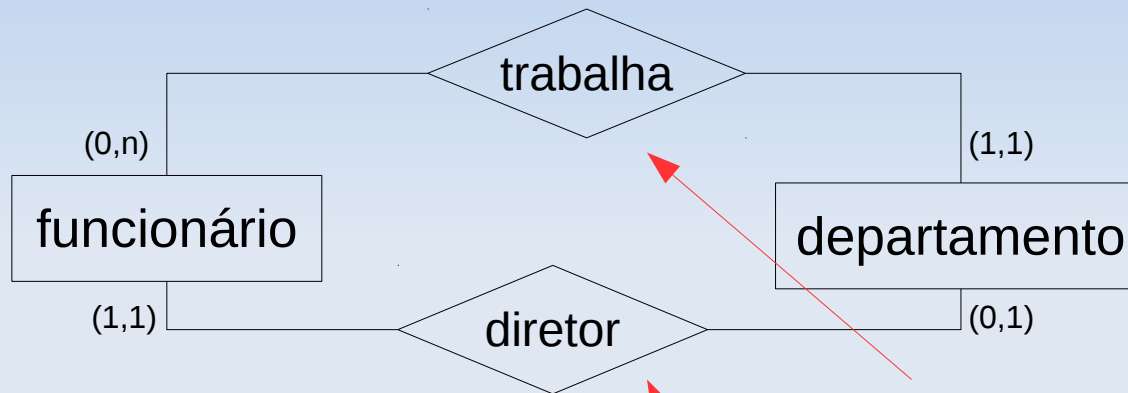
peca(cod, nome, prc, qtestoq)

compoe(codp(peca), codparte(peca), qtdade)



Conversão

- Relacionamentos múltiplos



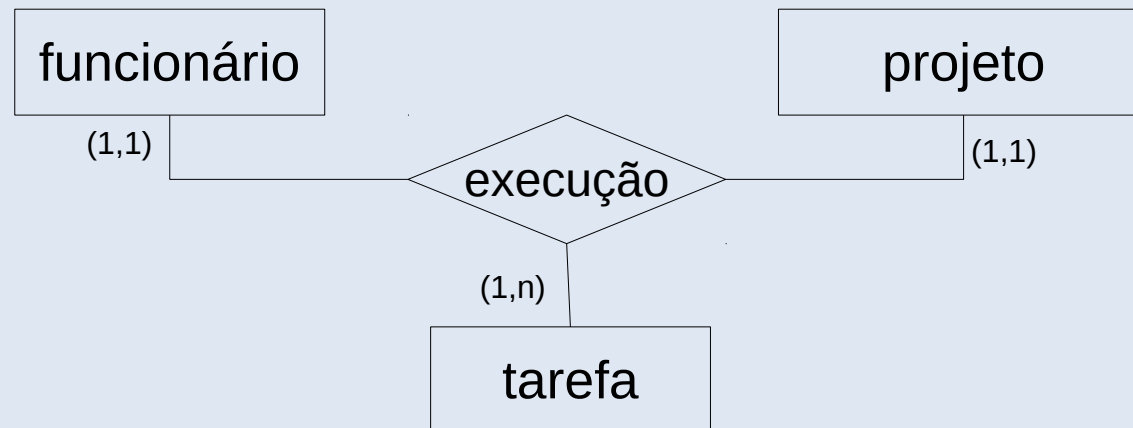
`func(mat, nome, dtadm, codd(depto))`

`depto(cod, nome, matd(func))`

Conversão

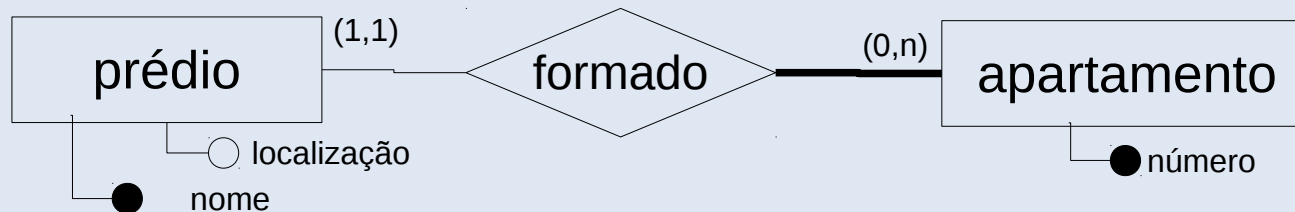
- Relacionamento Ternário
 - Geralmente, transforma a relação em tabela como em relacionamentos N:N
 - A PK é definida conforme a necessidade de cardinalidade
`execucao(matf(func), codp(prjt), codt(tarefa))`

PK a definir



Conversão

- Relacionamento identificador (entidade fraca)
 - Mesma estratégia de outros casos porém a chave estrangeira se torna primária também
predio (nome, local)
 - apto (nome(predio), nro)

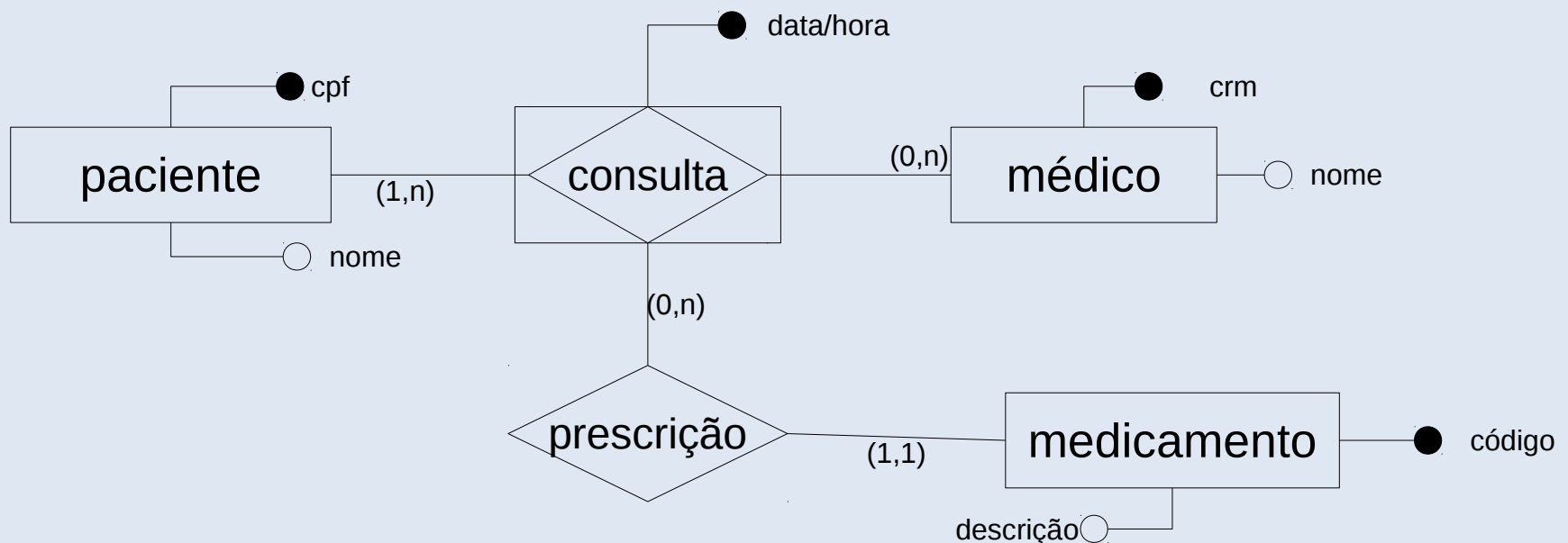


Conversão

- Entidade associativa

- Caso similar ao do relacionamento ternário: relacionamento se torna uma tabela

`consulta(crm(medico), cpf(paciente), dthora, codm(medic))`

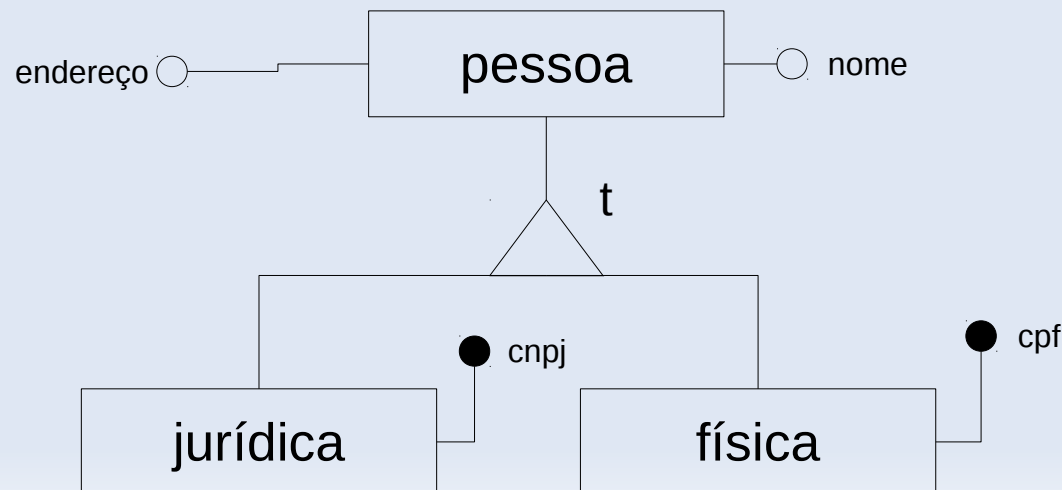


Conversão

- Especialização

- Caso total

- Cria-se tabelas apenas para as especializações
`pesj(cnpj, nome, ender)` `pesf(cpf, nome, ender)`
 - Cria-se apenas uma tabela
`pessoa(cod, nome, ender, cpf, cnpj, tipo)`



Conversão

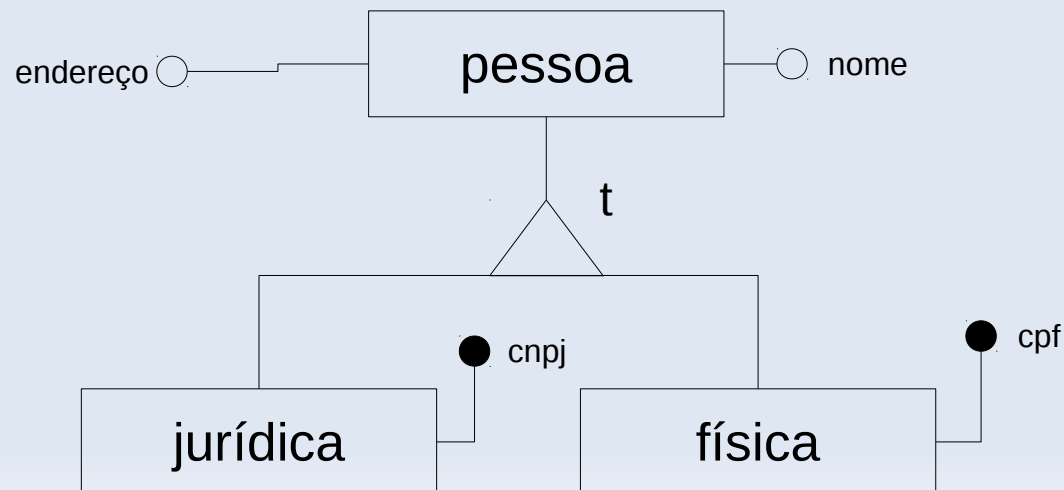
- Especialização

- Caso total

- Cria-se todas as tabelas

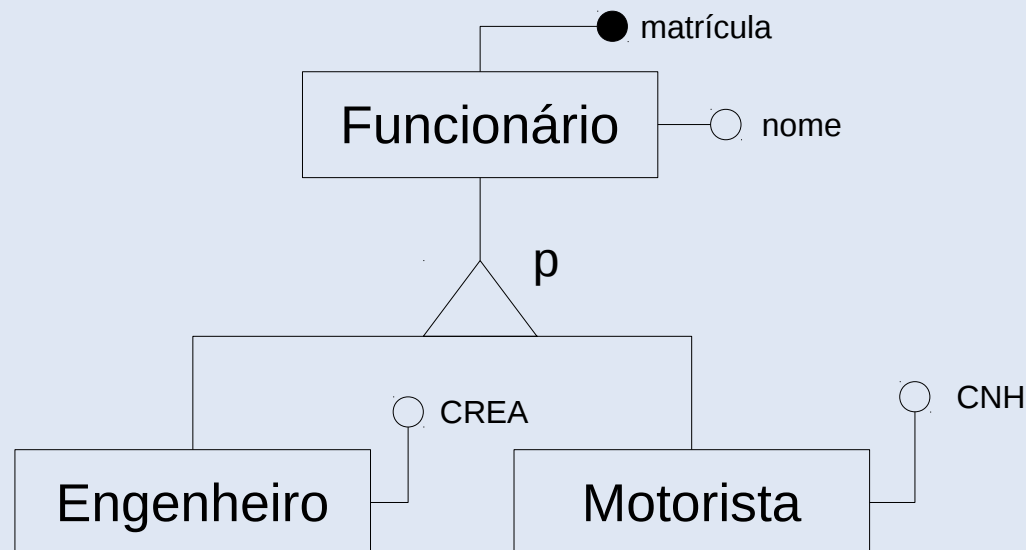
`pesj(cod(pessoa), cnpj)` `pesf(cod(pessoa), cpf)`

`pessoa(cod, nome, ender)`

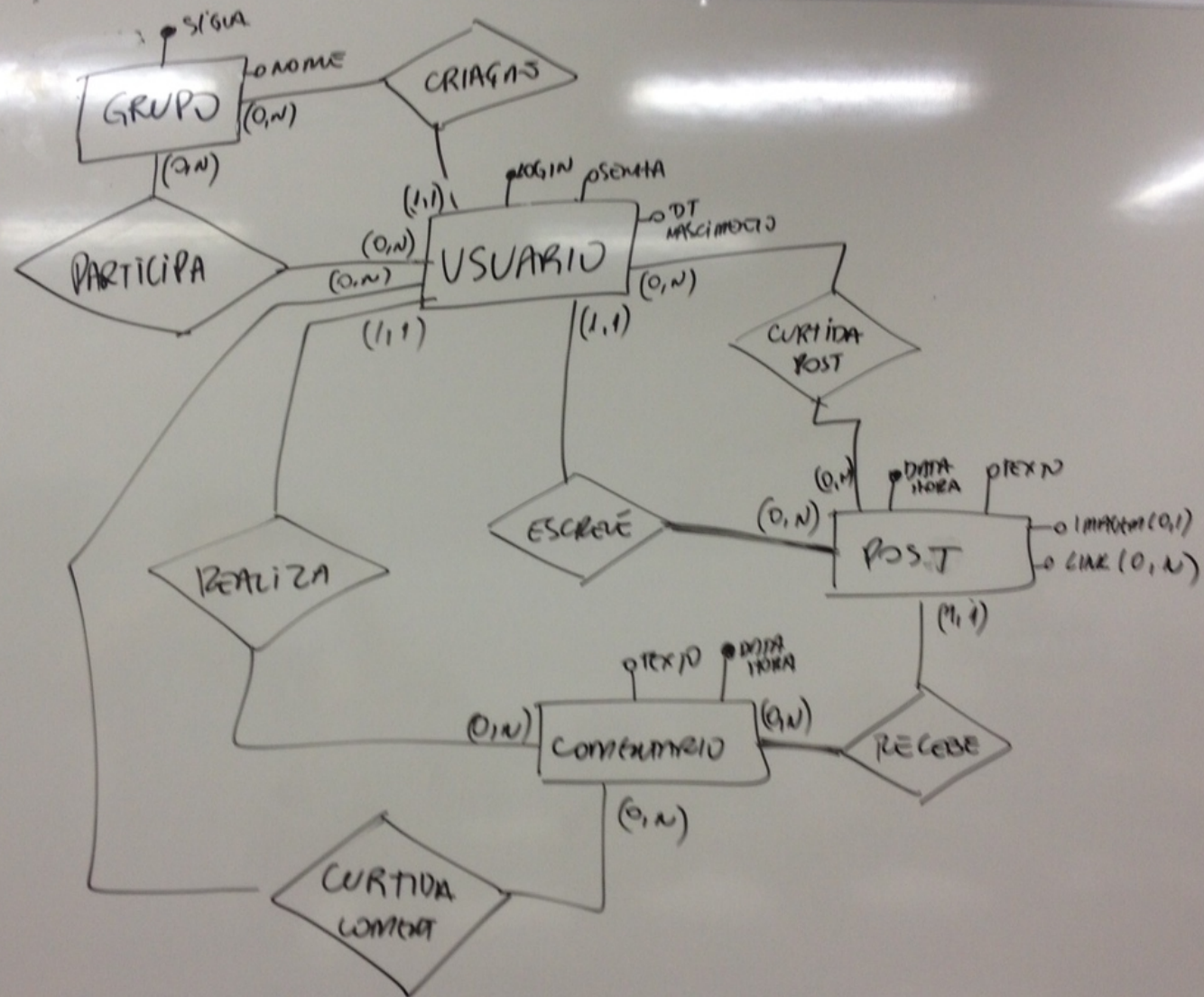


Conversão

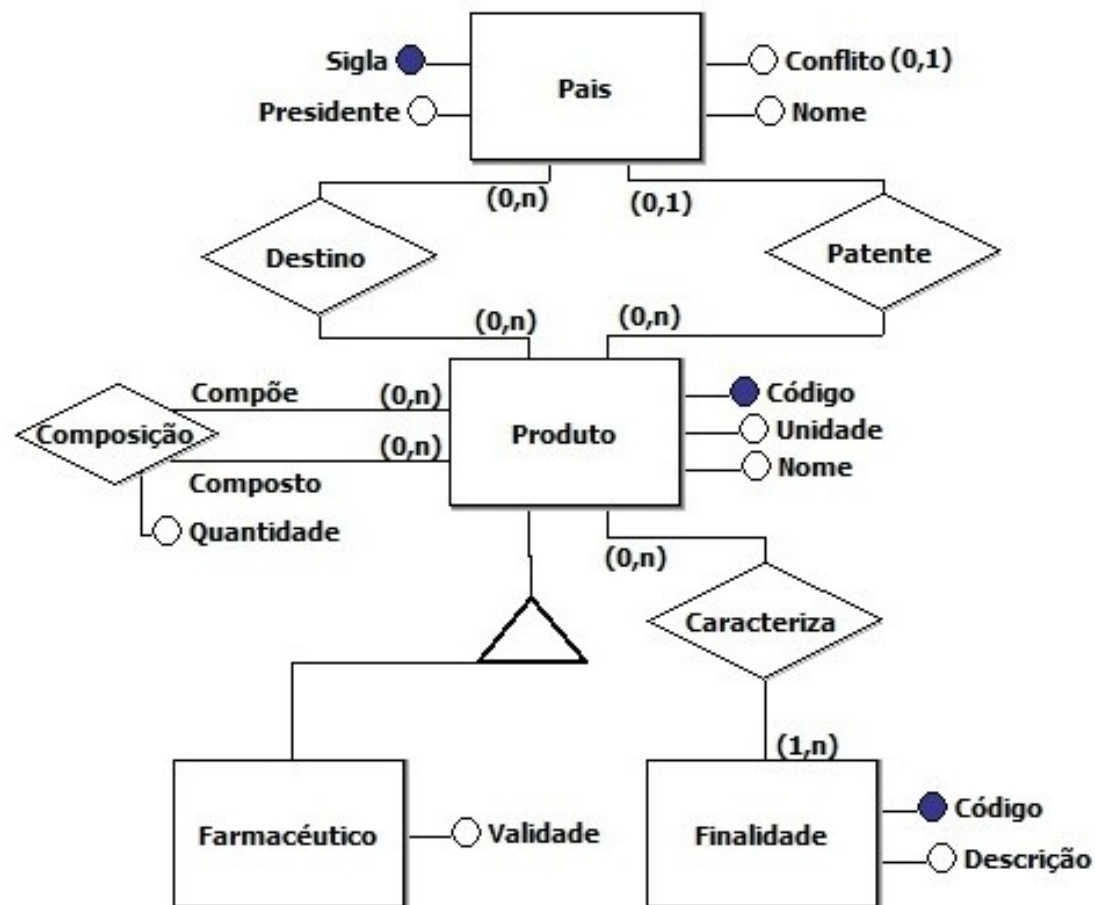
- Especialização
 - Caso Parcial
 - Casos similares ao total exceto que deve ser criada a tabela *mãe* sempre



Exercício 01



Exemplo



Exemplo

