

[cin.ufpe.br](http://cin.ufpe.br)



UNIVERSIDADE FEDERAL DE PERNAMBUCO



# Projeto Conceitual de BD

## (Conceitos Avançados - Parte 2)



Por: Prof. Robson do Nascimento Fidalgo  
[rdnf@cin.ufpe.br](mailto:rdnf@cin.ufpe.br)

# Peculiaridades do MER

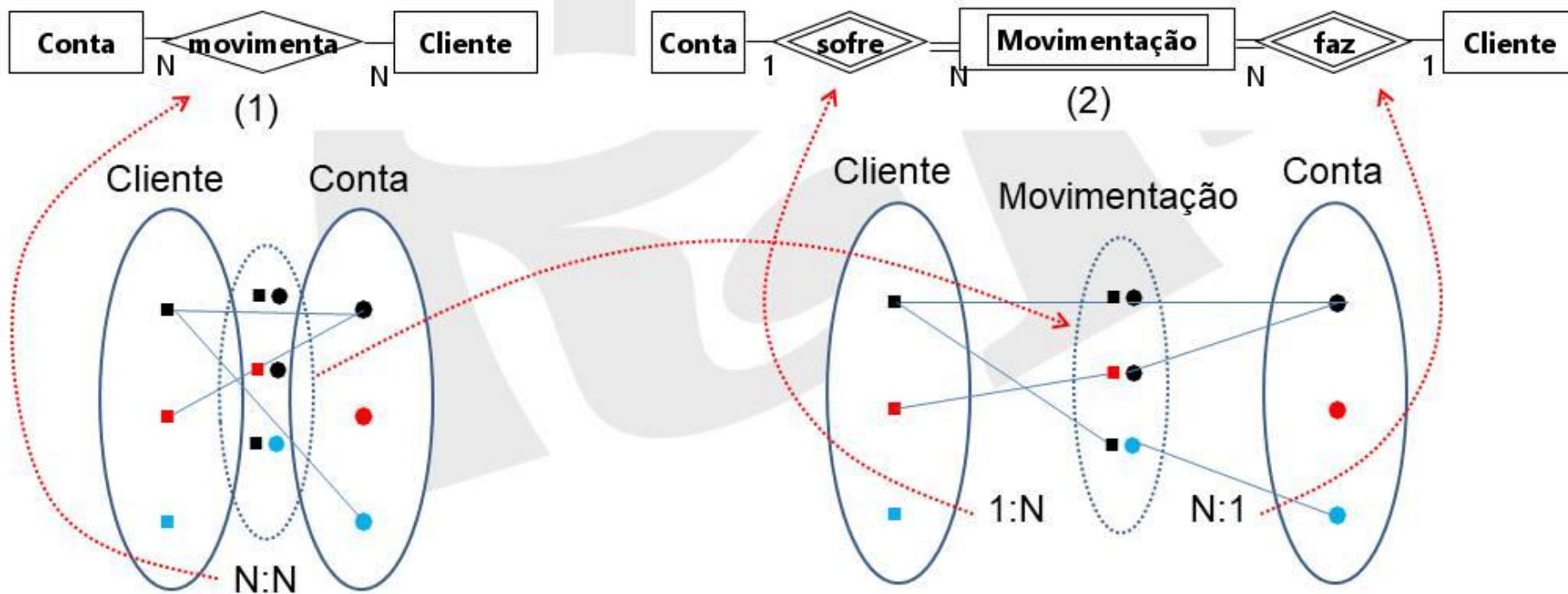
- ▶ Notação gráfica simples e não ambígua
  - ▶ Leitores devem entender exatamente a mesma coisa
  - ▶ É importante que todos os envolvidos estejam treinados
  - ▶ A sua simplicidade pode levar a sua sub-utilização, servindo apenas como ferramenta informal para apresentação de ideias

# Peculiaridades do MER

- ▶ Apresenta apenas uma visão conceitual da estrutura do BD
- ▶ Tem poder de expressão limitado
  - ▶ Valores válidos e pós/pré condições devem ser informadas à parte
  - ▶ EX.
    - Não pode haver saldo negativo
    - Não pode haver depósito com valor inferior a X
    - Nenhum salário pode ser reduzido
    - O salário de um empregado não pode ser superior ao do seu chefe
    - ....

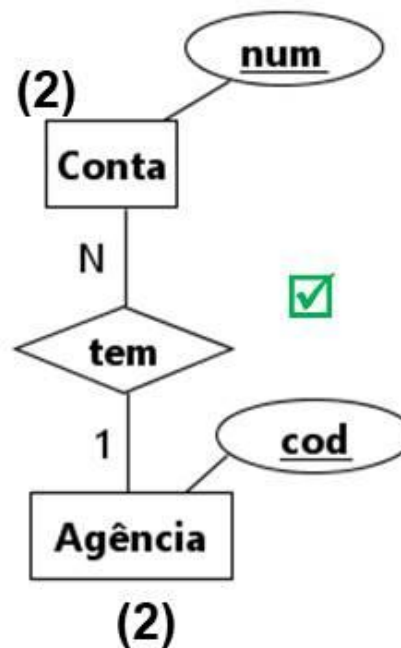
# Peculiaridades dos MER

- ▶ Esquemas ER diferentes podem ser equivalentes
  - ▶ Para fins de projeto de BD, dois esquemas ER são equivalentes se esses capturam a mesma realidade (geram o mesmo banco de dados)
  - ▶ Os esquemas 1 e 2 são visualmente diferentes, mas geram o mesmo banco de dados



# Dúvidas frequentes

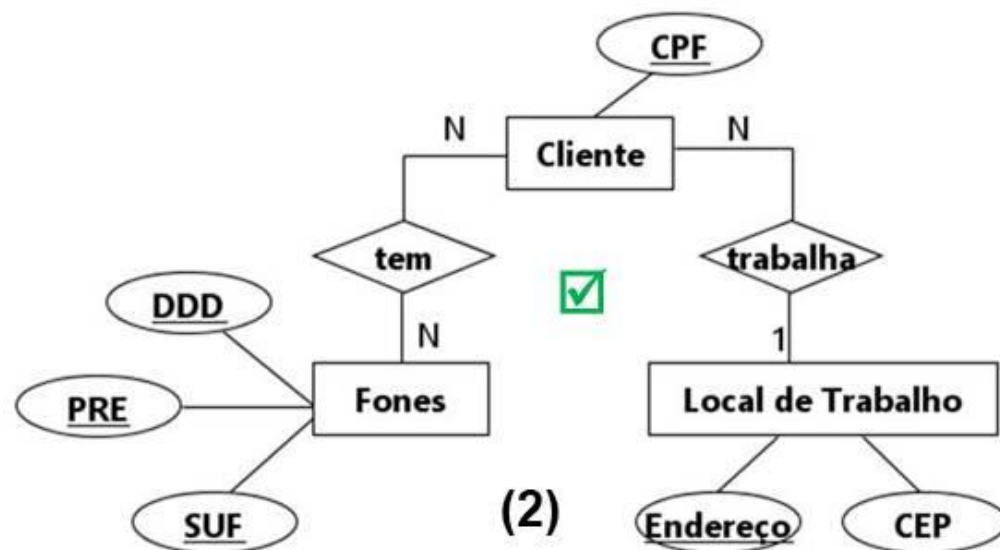
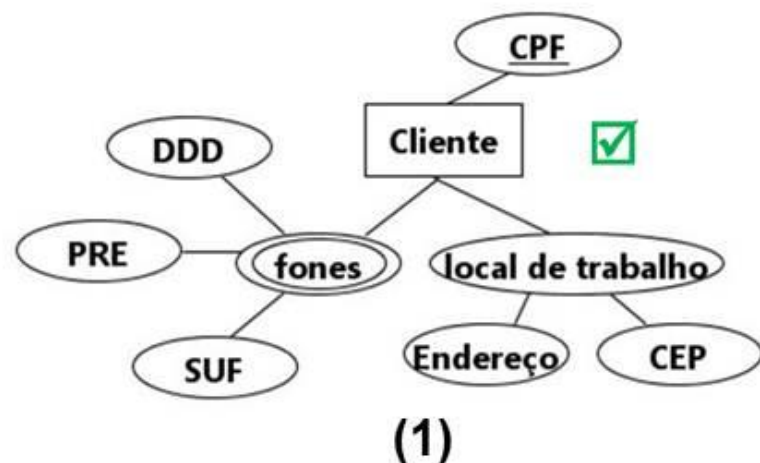
- ▶ Devo modelar um conceito como atributo ou entidade?
  - ▶ Se é meramente descritivo → atributo (esquema 1)
  - ▶ Se tem um identificador explícito → entidade (esquema 2)





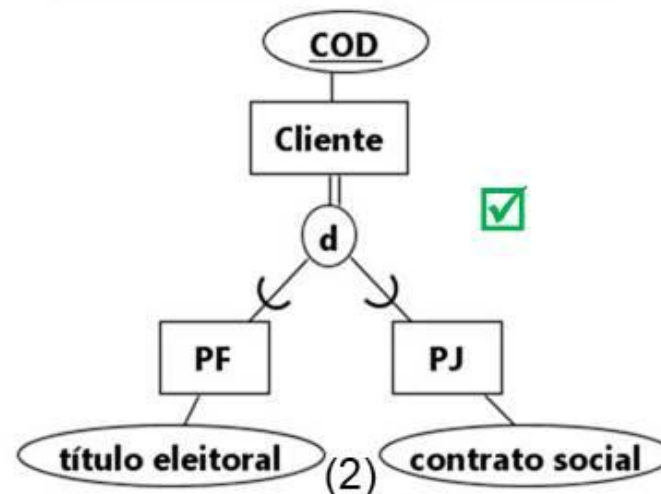
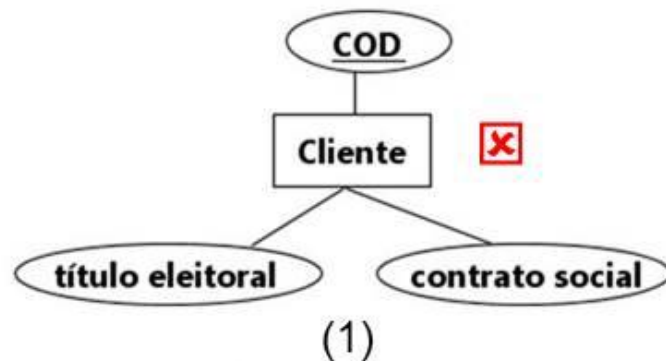
# Dúvidas frequentes

- Devo modelar um conceito que tem muitas propriedades como um atributo composto/multivalorado ou como uma entidade?
- Se é exclusivo de uma entidade → atributo (esquema 1)
- Se pode ser compartilhado entre várias entidades → entidade (esquema 2)



# Dúvidas frequentes

- ▶ Devo modelar conceitos como atributos ou usar herança?
  - ▶ Se pode haver inconsistência entre os conceitos → herança

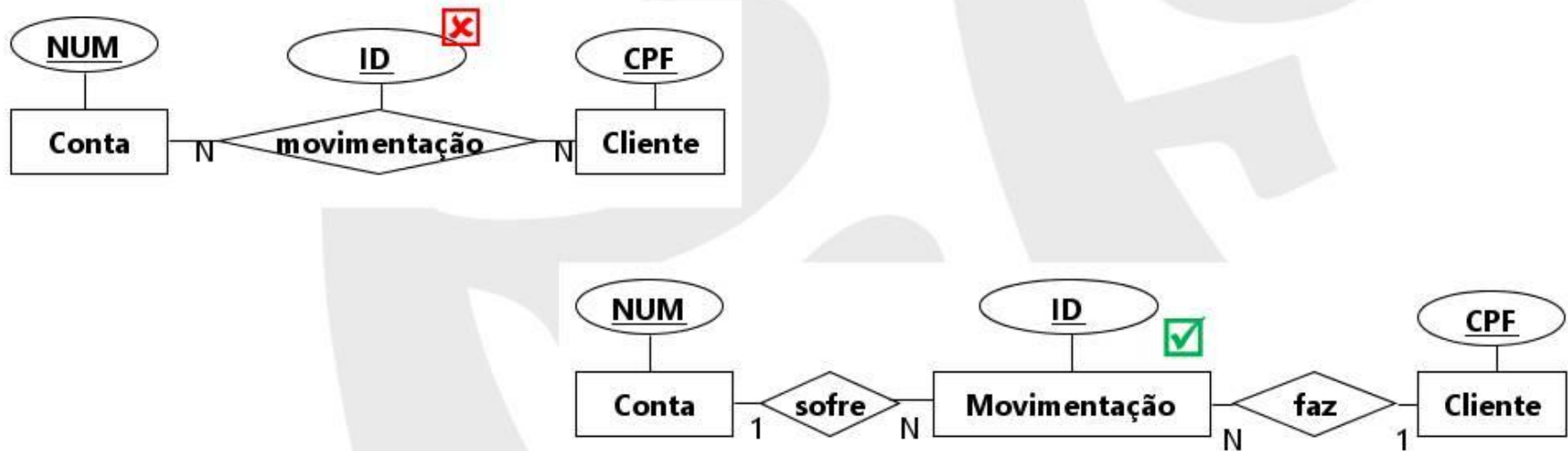


- ▶ No esquema 1, um cliente pode ter título eleitoral e contrato social ao mesmo tempo



# Dúvidas frequentes

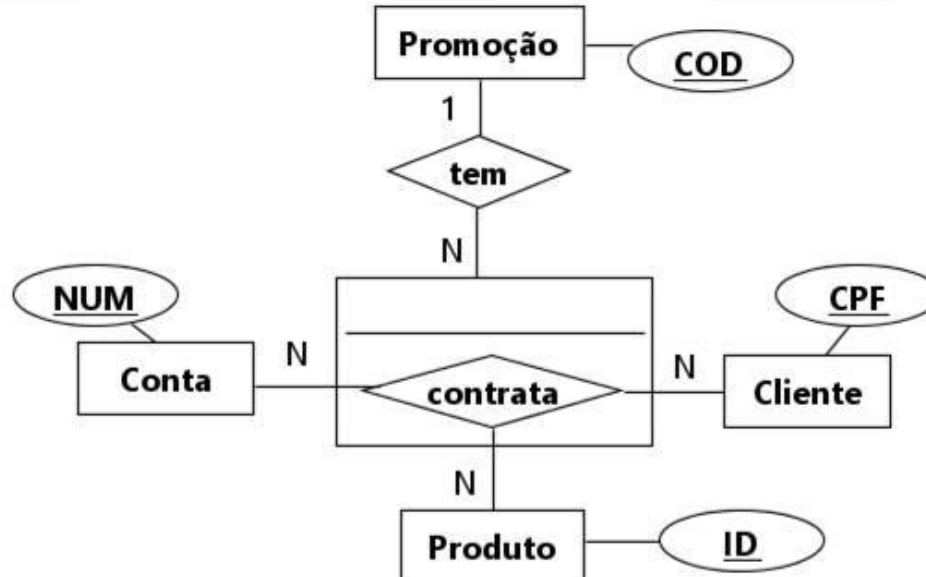
- ▶ Devo modelar um conceito como relacionamento ou Entidade?
- ▶ Se tem identificador explicito → entidade



Atenção: relacionamentos não podem ter identificadores!

# Dúvidas frequentes

- ▶ Devo modelar um conceito como relacionamento N-ário ou entidade associativa?
  - ▶ Se todas as vezes que o relacionamento ocorrer, esse SEMPRE envolve TODAS as entidades participantes → N-ário
    - ❑ No esquema abaixo, “contrata” sempre envolve Conta, Cliente e Produto, mas não precisa envolver Promoção



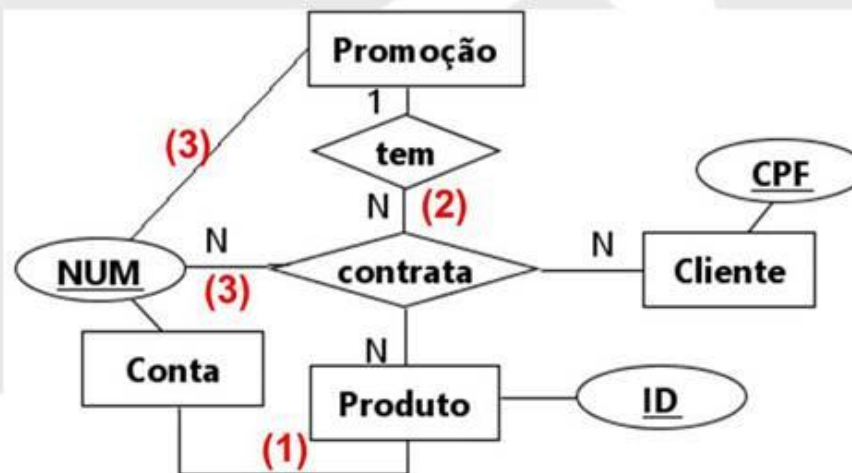
# Requisitos para ter um bom MER

- ▶ Ser sintaticamente correto
- ▶ Ser semanticamente correto
- ▶ Evitar ou controlar construções redundantes
- ▶ Capturar o aspecto temporal
- ▶ Ser completo



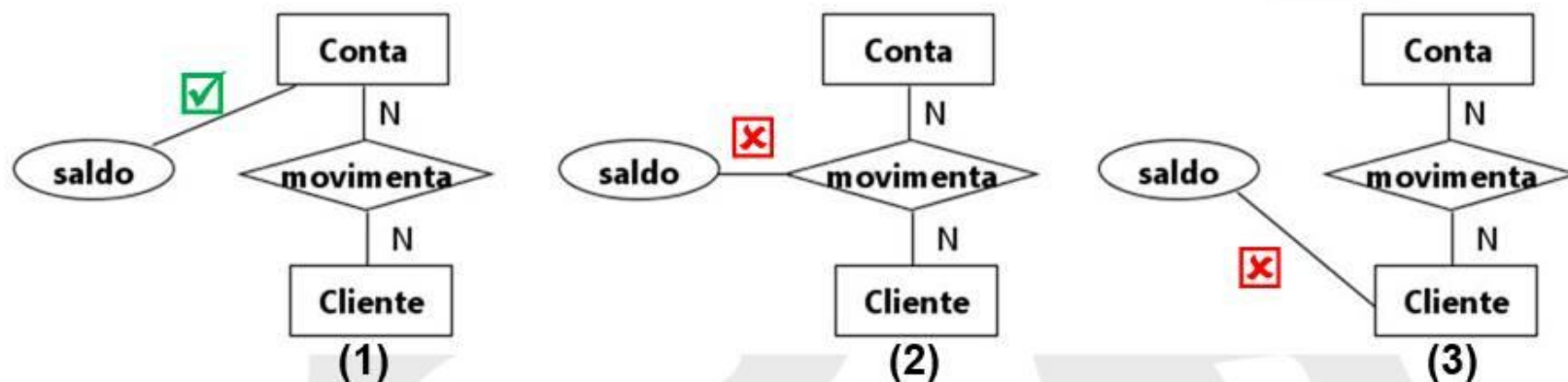
# Ser sintaticamente correto

- ▶ O esquema deve respeitar as regras sintáticas de construção
  - ▶ Erros sintáticos são evitados por boas ferramentas CASE
  - ▶ Exemplos de erros sintáticos:
    - (1) ligar duas entidades sem um relacionamento
    - (2) ligar dois relacionamentos sem usar uma entidade associativa
    - (3) ligar um atributo a mais de uma entidade ou relacionamento



# Ser semanticamente correto

- ▶ O esquema não pode ter atributos mal especificados

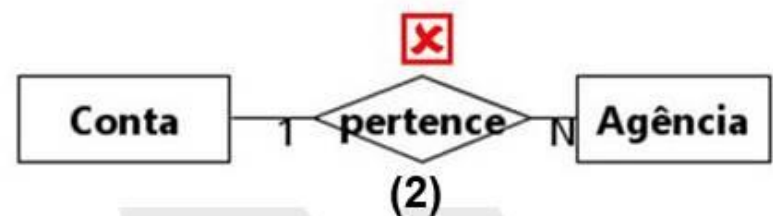
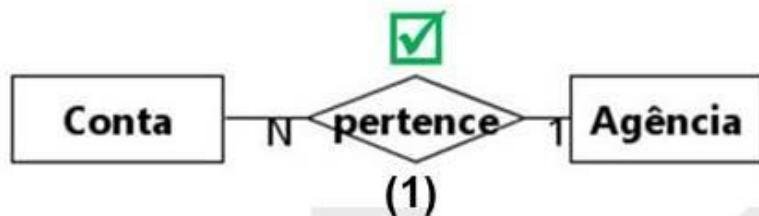


- ▶ No esquema 1, cada conta tem seu saldo e todos os cliente vêm o mesmo saldo
- ▶ No esquema 2, uma conta pode ter um saldo diferente para cada um dos seus clientes
- ▶ No esquema 3, um cliente terá o mesmo saldo para todas as suas contas



# Ser semanticamente correto

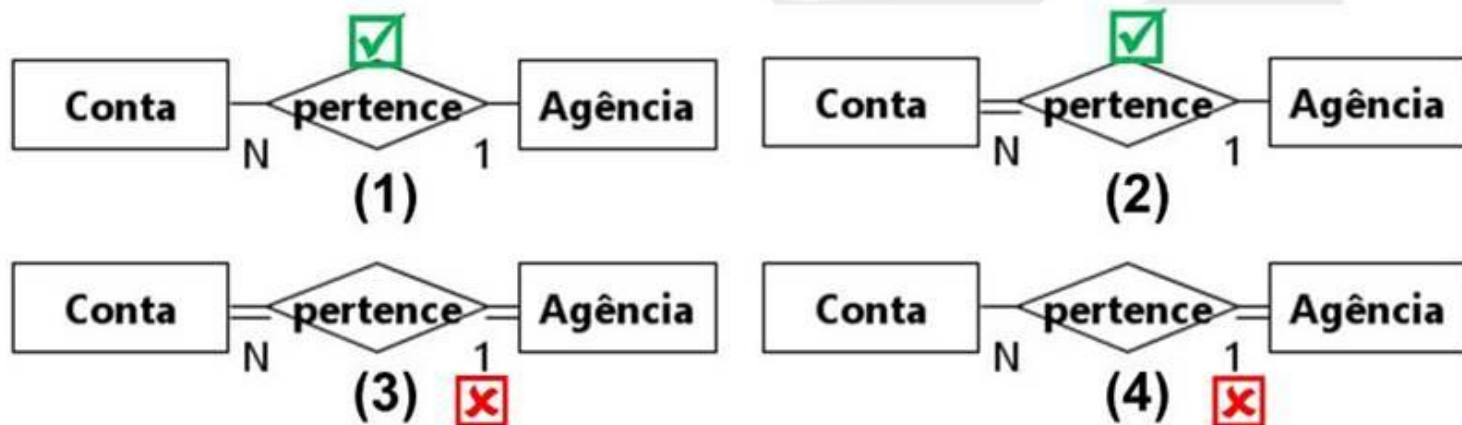
- ▶ O esquema não pode ter relacionamentos com cardinalidades mal especificadas



- ▶ No esquema 1, cada conta pertence a uma única agência, a qual pode ter várias contas
- ▶ No esquema 2, uma conta pode pertencer a várias agências, mas uma agência só pode ter uma conta

# Ser semanticamente correto

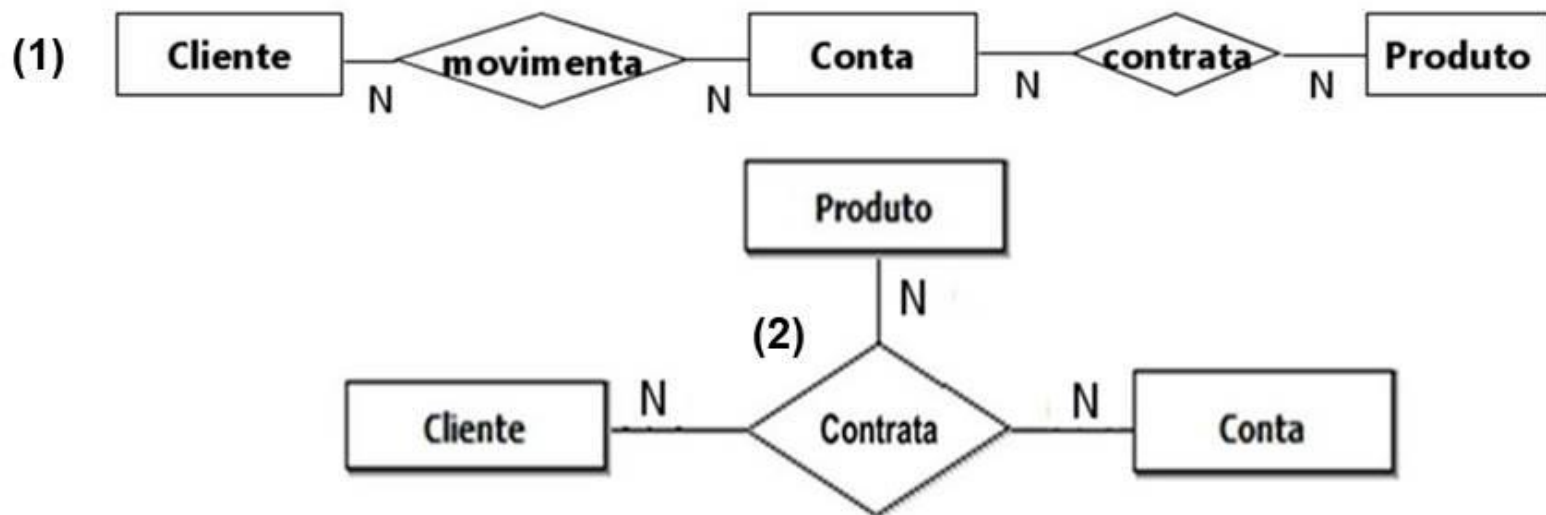
- ▶ O esquema não pode ter relacionamentos com participações mal especificadas



- ▶ O esquema 1 é flexível (aceitável)
- ▶ No esquema 2, toda conta deve ter uma agência para ser cadastrada (bom)
- ▶ O esquema 3 gera uma dependência mútua (desnecessário)
- ▶ No esquema 4, toda agência deve ter uma conta para ser cadastrada (estranho)

# Ser semanticamente correto

- ▶ O esquema não pode ter relacionamentos com grau mal especificados



- ▶ O esquema 1 garante que clientes não podem ter produtos diferentes na conta
- ▶ O esquema 2 permite que clientes tenham produtos diferentes na conta
- ▶ Os dois esquemas estão corretos (depende da realidade)

# Ser semanticamente correto

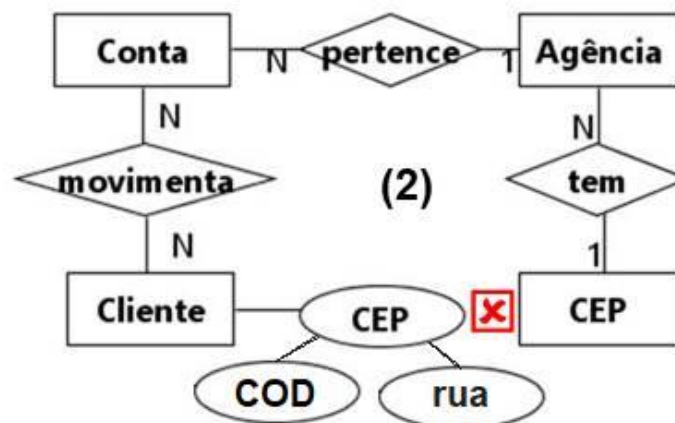
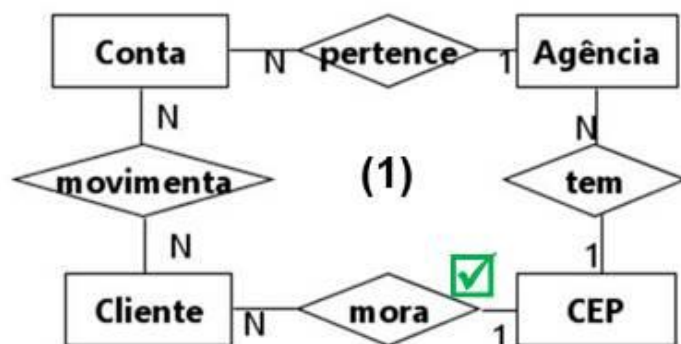
- ▶ O esquema não pode ter relacionamentos com grau mal especificados



- ▶ O esquema 1 garante que uma conta é de apenas uma agência
- ▶ O esquema 2 permite que uma conta exista em mais de uma agência

# Ser semanticamente correto

- ▶ O esquema não pode capturar mais de uma realidade

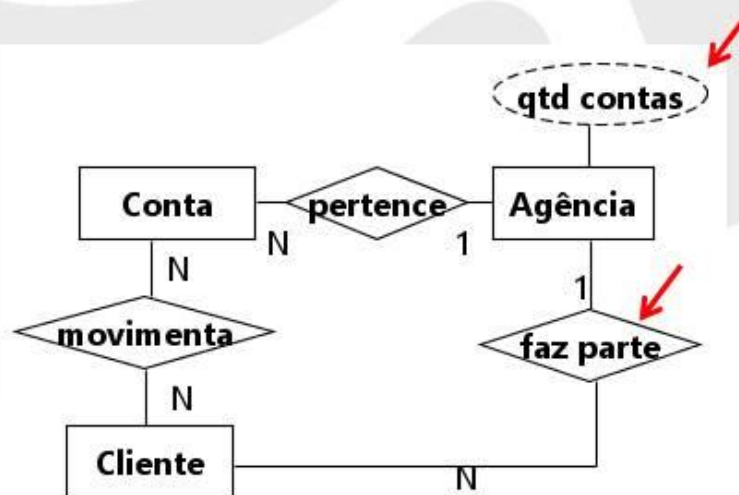


- ▶ O esquema 1 concentra tudo na entidade CEP, impedindo que haja inconsistências entre "COD" e "rua"
- ▶ O esquema 2 permite que uma "rua" tenha mais de um "COD" e vice-versa



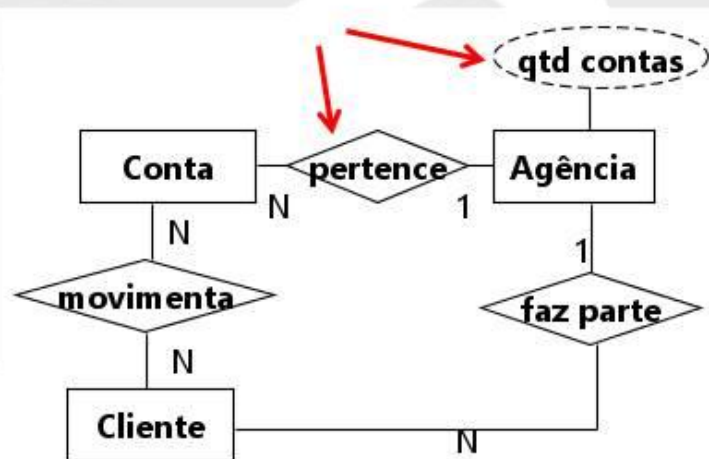
# Evitar ou controlar construções redundantes

- ▶ Construções redundantes podem melhorar o desempenho, mas também podem gerar dados inconsistentes
- ▶ Evite ou controle as construções redundantes
  - ▶ “qtd contas” e “faz parte” são construções redundantes (se não forem controladas via programação podem gerar dados contraditórios)



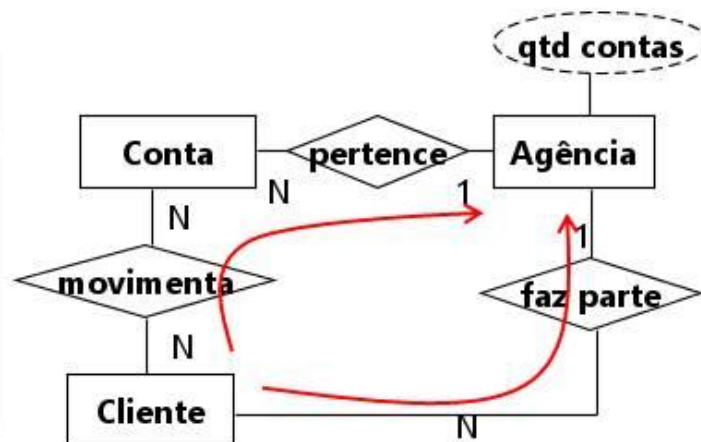
# Evitar ou controlar construções redundantes

- ▶ Saber a quantidade de contas de uma agência via o atributo “qtd contas” é mais rápido via o relacionamento “pertence”
- ▶ Para evitar inconsistência, deve-se programar uma rotina para atualizar o atributo “qtd contas” toda as vezes que uma conta for criada, excluída ou trocada de agência (avaliar custo X benefício de “qtd contas”)



# Evitar ou controlar construções redundantes

- ▶ Via “faz parte”, é mais rápido saber a agência de um cliente, mas “faz parte” pode registrar uma agência cujo cliente não tem conta
- ▶ Para evitar inconsistência, também deve-se programar uma rotina para impedir a contradição citada acima (avaliar custo X benefício de “faz parte”)



# Capturar o aspecto temporal

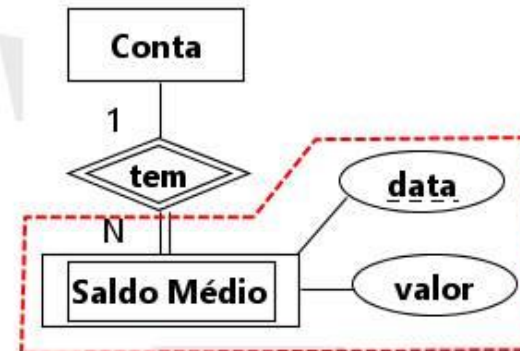
- ▶ Certas aplicações precisam que o BD guarde o histórico de alguns atributos ou relacionamentos
  - ▶ Só guardar o histórico quando for necessário, pois é custoso
- ▶ Fazer isso exige usar um selo temporal (uma data ou um instante de tempo) para registrar o acontecimento do fato

# Capturar o aspecto temporal

## ► Guardando o histórico de um atributo



(1)



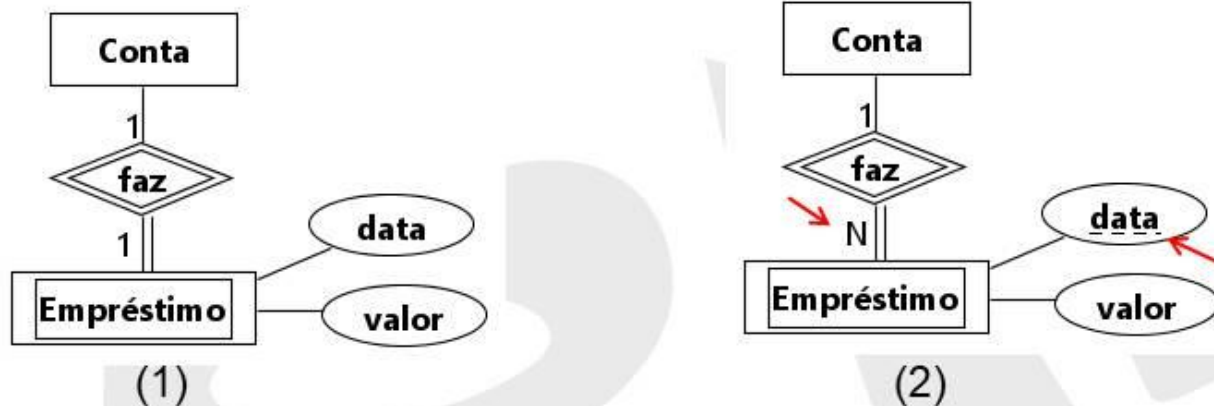
(2)

- O esquema 1 guarda apenas o saldo médio atual
- O esquema 2 guarda o histórico dos saldos médios



# Capturar o aspecto temporal

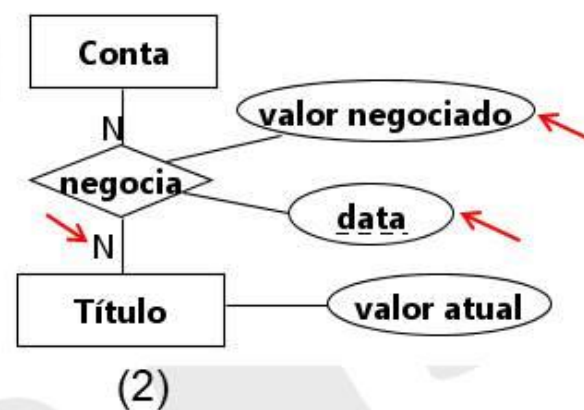
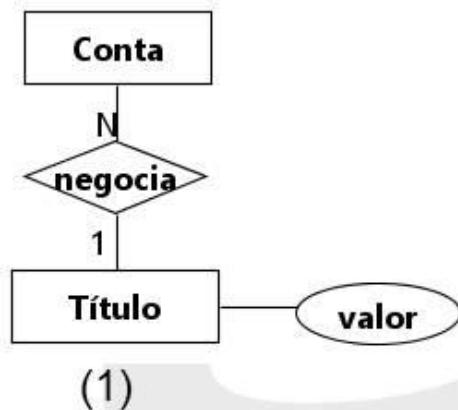
- Guardando o histórico de um relacionamento 1:1



- O esquema 1 guarda apenas os dados do empréstimo atual
  - Ao longo do tempo uma conta pode fazer vários empréstimos, mas apenas o atual é guardado (o atual sobrepõe o anterior)
- O esquema 2 guarda o histórico dos empréstimos
  - Pode-se usar um instante (em milissegundos) no lugar da data

# Capturar o aspecto temporal

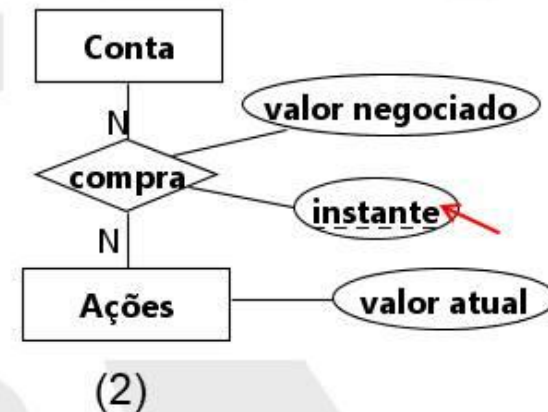
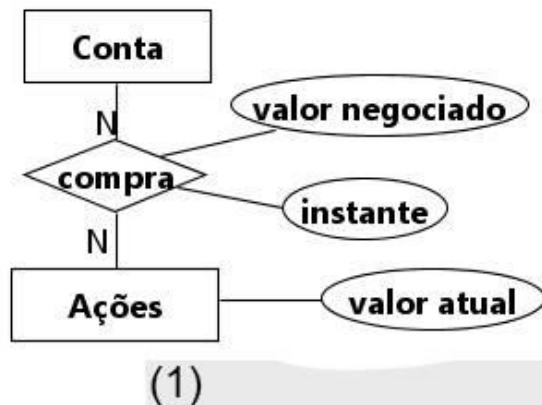
- Guardando o histórico de um relacionamento 1:N



- O esquema 1 guarda apenas os dados do título atual
- O esquema 2 guarda o histórico dos títulos negociados, incluindo se o cliente negociar mais de uma vez o mesmo título

# Capturar o aspecto temporal

- Guardando o histórico de um relacionamento M:N



- O esquema 1 guarda quase todo o histórico das ações compradas
- O esquema 2 estende o esquema 1, pois consegue registrar as recompras de uma mesma ação

# Ser completo

- ▶ Garantir que o esquema atende aos requisitos do BD
- ▶ Deve-se envolver fortemente o usuário
  - ▶ Peça para os usuários fazerem uma lista com as restrições e consultas mais importantes
  - ▶ Faça reuniões até não haver mais nenhuma dúvida

Atenção: É mais fácil e barato corrigir um erro no projeto conceitual do que em qualquer outra fase de projeto do BD



# Estratégias de Modelagem

- ▶ O projeto conceitual do BD pode ser feito a partir:
  - ▶ de Informações existentes
    - ▶ Estratégia Engenharia Reversa (feita a partir de sistemas)
      - Feita automaticamente por uma ferramenta CASE
    - ▶ Estratégia Bottom-up (feita a partir de formulários e documentos)
      - (1º) Atributos → (2º) Entidades → (3º) Relacionamentos → (4º) Herança
  - ▶ do conhecimento de especialistas
    - ▶ Estratégia Top-down
      - (1º) Entidades → (2º) Relacionamentos → (3º) Heranças → (4º) Atributos
    - ▶ Estratégia Inside-out
      - igual ao top-down, mas começa pelas entidades mais importantes



[cin.ufpe.br](http://cin.ufpe.br)



UNIVERSIDADE FEDERAL DE PERNAMBUCO