

QUESTIONÁRIO 3

Atributo exigir criar nova relação

1. O mapeamento de atributos que são apenas compostos nunca exige a criação de uma nova relação.
2. O mapeamento de atributos que são apenas multivalorados nunca exige a criação de uma nova relação.
3. O mapeamento de atributos que são apenas derivados nunca exige a criação de uma nova relação.
4. O mapeamento de atributos que são apenas discriminadores nunca exige a criação de uma nova relação.
5. O mapeamento de atributos que são apenas identificadores nunca exige a criação de uma nova relação.

Apenas se for multivalorado! Os outros NÃO exigem a criação de nova relação, só se for outra coisa e multivalorado. Exige pois pode aceitar muitos valores, mas em casos que a quantidade é conhecida e pequena é mais vantajoso criar novos campos/colunas.

Viável para qualquer tipo de herança

6. O mapeamento que cria uma relação para cada entidade da herança é viável para qualquer tipo de herança.

Porque quando se cria uma relação pra cada entidade, a superentidade e as subentidades, a implementação permite todos os tipos pq sempre vai ser feita a junção entre as sub e a super.

7. O mapeamento que apenas cria uma relação para cada subentidade da herança é viável para qualquer tipo de herança.

Não abrange a herança parcial! Pois na parcial é quando tem instância na super e não tem na sub. Como só mapeia as subs, não vai ter como capturar as instâncias das superentidades.

8. O mapeamento que cria uma única relação para toda a herança é viável para qualquer tipo de herança.

É viável, mas o custo é alto por causa das junções e fica uma gambiarra.

Atributos opcionais

9. O mapeamento que cria uma relação para cada entidade da herança gera atributos opcionais.

Elas já vão englobar tudo aquilo que aquela entidade vai ter, sem a necessidade de atributos opcionais.

10. O mapeamento que apenas cria uma relação para cada subentidade da herança gera atributos opcionais.

Não gera, porque não juntou nada então vai ter apenas os da sub com o da super. E pode até herdar opcionais da super, mas não gera os atributos opcionais.

11. O mapeamento que cria uma única relação para toda a herança gera atributos opcionais.

Como cria uma relação para toda a herança, as subentidades podem ter muitos atributos, e isso pode ter muitos opcionais na hora de juntar, porque não necessariamente vai usar todos eles.

Exige junções

12. O mapeamento que cria uma relação para cada entidade da herança exige junções quando é necessário obter simultaneamente dados das relações que mapeiam a superentidade e as subentidades.

Como mapeou cada entidade da herança é necessário que a junção pegue os dados simultâneos da super e das subentidades para ser colocada na relação de maneira correta.

Ex: a chave primária da super migra como primária e estrangeira para as subs → mapeamento 1:1, uma instância da super mapeada equivale a uma da sub

13. O mapeamento que cria relações apenas para as subentidade da herança exige junções quando é necessário obter simultaneamente dados das relações que mapeiam a superentidade e as subentidades.

Não usa junções porque só mapeia para a subentidade, então tudo que está na super migra para a sub e não precisa usar junção!

14. O mapeamento que cria uma única relação para toda a herança exige junções quando é necessário obter simultaneamente dados das relações que mapeiam a superentidade e as subentidades.

Como cria uma relação só contendo tudo não usa junção, já que vai ser tudo englobado na relação.

Exige que a PK seja FK

15. O mapeamento que cria uma relação para cada entidade da herança exige que as PK das relações que mapeiam as subentidades também sejam FK para a relação que mapeia a superentidade.

Porque é necessário que a chave primária da super migre como primária, mas também deve ser estrangeira para as subentidades, para fazer a ligação entre essas entidades na relação.

16. O mapeamento que apenas cria uma relação para cada subentidade da herança exige que as PK das relações que mapeiam as subentidades também sejam FK para a relação que mapeia a superentidade.

Como você mapeia só as subentidades não vai ter chave estrangeira apontando para a superentidade. Já que não tem super não tem como pegar a primária dela, é como se a super não existisse.

Programação de rotinas

17. O mapeamento que cria uma relação para cada entidade da herança exige a programação de rotinas para garantir a qualidade dos dados.

Não há junção de relações, pois é um mapeamento para cada entidade da herança, então já evita que seja necessário fazer a avaliação dos dados.

18. O mapeamento que apenas cria uma relação para cada subentidade da herança exige a programação de rotinas para garantir a qualidade dos dados.

Esse mapeamento já evita que seja necessário fazer a avaliação dos dados, já que esses dados já estão separados, cada um em sua respectiva subentidade, se for herança total.

19. O mapeamento que cria uma única relação para toda a herança exige a programação de rotinas para garantir a qualidade dos dados.

Como está tudo junto então precisa implementar muita coisa, e isso faz com que seja necessário as rotinas para garantir a qualidade dos dados que estão sendo colocados na relação.

FK formem a PK

20. O mapeamento de uma entidade associativa sobre um relacionamento M:N exige que as FK formem a PK da relação mapeada.

Como é M:N o relacionamento deve ter a chave do lado M e a chave do lado N formando a chave primária composta da relação. Mas cada campo separadamente da chave composta é estrangeira para suas respectivas tabelas.

Duas relações ligadas/fusionadas

21. Duas relações ligadas via a adição de uma chave estrangeira não precisam de junção quando é necessário acessar simultaneamente dados dessas duas relações.

Adição de chave estrangeira exige junção, já que está pegando uma chave e colocando em outra tabela. Para fazer o acesso simultâneo a essas duas relações é necessário usar a junção para acessar os dados certos.

22. Duas relações ligadas via a criação de uma terceira relação precisam de apenas uma junção quando é necessário acessar simultaneamente os dados das duas primeiras relações.

Precisa de 2 junções, porque se tem duas relações e cria uma terceira relação vai precisar de duas junções para poder fazer a consulta entre a terceira e a primeira, e entre a terceira e a segunda.

23. Duas relações fusionadas eventualmente precisam de junção quando é necessário acessar simultaneamente dados dessas duas relações.

Não precisa de junção pois a fusão junta tudo em uma única tabela. A fusão é feita justamente para evitar a junção.

Exigência e custo

24. Duas relações ligadas via a criação de uma relação sempre exigem mais junções do que qualquer alternativa quando é necessário acessar simultaneamente dados dessas duas relações.

25. Duas relações ligadas via a adição de uma chave estrangeira sempre exigem menos junções do que qualquer alternativa quando é necessário acessar simultaneamente dados dessas duas relações.

26. Duas relações fusionadas sempre exigem menos junções do que qualquer alternativa quando é necessário acessar simultaneamente dados dessas duas relações.

São 3 possibilidades de juntar: a fusão-relação, que não tem junção, a adição de chave estrangeira, que tem 1 junção, e a criação de uma nova relação que faz necessário a adição de 2 junções. Sempre que forem criadas mais tabelas vai criar mais junções.

27. A criação de uma relação é a alternativa mais indicada para mapear relacionamentos 1:1.

Relacionamento 1:1 pode fazer fusão, adicionar chave estrangeira ou nova relação, mas a mais custosa é a criação de nova relação, então ela não é a mais indicada.

28. A fusão de relações é a alternativa mais indicada para mapear relacionamentos M:N.

Como não é possível implementar relacionamento M:N com fusão, então não vai ser indicada em nenhum momento para mapear esses relacionamentos.

29. A adição de uma chave estrangeira é a alternativa mais indicada para mapear relacionamentos 1:N.

Sim, a adição é a mais indicada, já que não pode fusão e uma nova relação é muito custosa para esse tipo de relacionamento.

1:1 total-total

30. No mapeamento de um relacionamento 1:1 total-total via a fusão de relações, a PK da relação fusionada pode ser qualquer uma das PK das relações originárias.

Sim, é uma escolha arbitrária. O que não pode acontecer é ter as duas chaves funcionando como primária de maneira composta, é ou uma ou outra.

31. No mapeamento de um relacionamento 1:1 total-total via a fusão de relações, a PK da relação fusionada nunca pode ser a composição das PK das relações originárias.

Não pode ter uma chave primária composta, já que isso poderia gerar contradição lógica e quebrar a cardinalidade da relação.

32. No mapeamento de um relacionamento 1:1 total-total via a fusão de relações, nunca se gera uma FK, mas sempre se gera uma AK obrigatória.

Porque como fez a fusão, não tem chave estrangeira, já que seria uma tabela só. Como a chave primária não pode ser composta, só um dos lados será a chave primária, o que gera a chave alternativa do outro lado.

1:1 parcial-parcial

33. No mapeamento de um relacionamento 1:1 parcial-parcial via a adição de uma chave estrangeira, a FK pode ser qualquer uma das PK das relações envolvidas.

Assim como na total-total, é uma escolha arbitrária. Como é parcial eu só preciso escolher uma, a outra ligação por ser parcial não se faz obrigatório e já vira opcional.

34. No mapeamento de um relacionamento 1:1 parcial-parcial via a adição de uma chave estrangeira, a FK nunca pode ser a composição das PK das relações envolvidas.

Assim como na total-total, não pode ter uma chave primária composta, já que isso poderia gerar contradição lógica e quebrar a cardinalidade da relação. Ele ser parcial-parcial não muda muita coisa pois ainda mantém o 1:1, logo, escolher a composição das PK quebra o relacionamento 1:1 formado.

35. No mapeamento de um relacionamento 1:1 parcial-parcial via a adição de uma chave estrangeira, a FK sempre é uma AK.

A AK é uma chave que poderia ser primária, visto que tem todas as particularidades pra ser, mas não foi a escolhida, assim como as chaves estrangeiras que poderiam ser primárias, mas são usadas para unir duas tabelas, visto que só escolhe uma chave pra ser a primária.

1:1 parcial-total

36. No mapeamento de um relacionamento 1:1 parcial-total via a adição de uma chave estrangeira, é melhor que a FK migre da relação com participação opcional para a relação com participação total.

Quando chega na total pela parcial ela vai chegar e vai ser única e obrigatória. Caso faça o contrário, ir da total para parcial vai ter que tornar a chave opcional, o que pode gerar problemas no mapeamento.

37. No mapeamento de um relacionamento 1:1 parcial-total via a adição de uma chave estrangeira, quando a FK migra da relação com participação opcional para a relação com participação total, essa FK deve ser obrigatória.

Porque como ela está indo para o lado com participação total, a FK chega como obrigatória já que para cadastrar o outro lado é preciso desse lado que ela migrou.

38. O mapeamento de um relacionamento 1:1 parcial-total via a adição de uma chave estrangeira exige que a FK seja uma AK.

Como ela vai pro lado total como única e obrigatória cabe a estrangeira virar uma alternativa, já que toda vez que é 1:1 a estrangeira deve ser uma alternativa, visto que pode ser primária ou não. mesmo ocorre no contrário pois só altera a obrigatoriedade. Toda vez que é 1:1 a estrangeira DEVE ser uma alternativa (só falta ver se é obrigatória ou não).

1:N parcial-parcial

39. O mapeamento de um relacionamento 1:N parcial-parcial via a adição de uma chave estrangeira permite que a FK migre de qualquer um dos lados.

A chave estrangeira que migra é a do lado 1. Já que a estrangeira SEMPRE vem do lado 1 do relacionamento porque é esse lado que identifica.

40. O mapeamento de um relacionamento 1:N parcial-parcial via a adição de uma chave estrangeira não exige que a FK seja obrigatória.

Só é obrigatória caso o lado N seja com participação total, aí acrescentaria a exclamação (!) na chave estrangeira. Mas, como é parcial-parcial não vai exigir a obrigatoriedade.

41. O mapeamento de um relacionamento 1:N parcial-parcial via a adição de uma chave estrangeira não exige que a FK seja uma AK.

Ter um relacionamento 1:N e falar que a estrangeira tem que ser alternativa insinua que o relacionamento é 1:1, o que gera contradição.

1:N parcial-total

42. O mapeamento de um relacionamento 1:N parcial-total via a adição de uma chave estrangeira exige que a FK migre do lado N para o lado 1.

A chave sempre migra do lado 1 para o lado N, independente da participação que tiver no relacionamento. Então NUNCA ocorre de N para 1.

43. No mapeamento de um relacionamento 1:N parcial-total via a adição de uma chave estrangeira, quando a FK migra da relação com participação opcional para a relação com participação total, essa FK deve ser obrigatória.

Quando ela vai da participação opcional para a participação total ela vai como obrigatória, pois o lado N será o lado total. Quando migra do lado 1 para o lado N (que seria total) vai migrar como obrigatória, porque depende da existência daquele lado.

44. O mapeamento de um relacionamento 1:N parcial-total via a adição de uma chave estrangeira exige que a FK seja uma AK.

Se colocar a chave estrangeira como alternativa não vai ser mais relacionamento 1:N e sim 1:1, o que geraria uma contradição lógica.

M:N

45. O mapeamento de um relacionamento M:N parcial-parcial via a criação de uma relação não exige que as FK sejam AK.

As estrangeiras quando é M:N não podem ser únicas. A composição delas formam a chave primária composta, mas quando estão sozinhas cada campo não pode ser uma chave alternativa.

46. O mapeamento de um relacionamento M:N parcial-total via a criação de uma relação exige que as FK sejam AK.

Se é M:N elas NÃO podem ser alternativas em NENHUM momento, se não gera uma contradição lógica.

47. O mapeamento de um relacionamento M:N total-total via a criação de uma relação exige que as FK sejam AK.

Se é M:N elas NÃO podem ser alternativas em NENHUM momento, se não gera uma contradição lógica entre os modelos.

48. O mapeamento de um relacionamento M:N parcial-parcial via a criação de uma relação exige que as FK formem a PK da relação criada.

Sim, pois a primária da relação entre as duas vai ser formada pela composição das chaves estrangeiras das duas outras relações antes de mapear, ou seja, vai ser uma chave primária composta. E cada estrangeira NÃO pode ser única!

49. O mapeamento de um relacionamento M:N parcial-total via a criação de uma relação exige que apenas a FK do lado total forme a PK da relação criada.

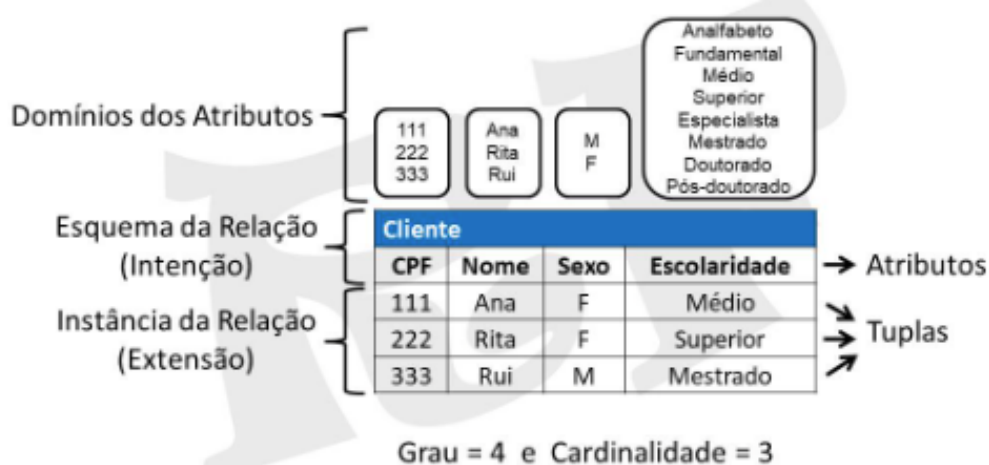
Todo relacionamento M:N SEMPRE a primária é composta formada pelas duas estrangeiras independente da participação que tem no relacionamento.

50. O mapeamento de um relacionamento M:N total-total via a criação de uma relação permite qualquer uma das FK forme a PK da relação criada.

Não é qualquer uma que DEVE formar, as duas irão se juntar e formar uma chave composta como PK da relação.

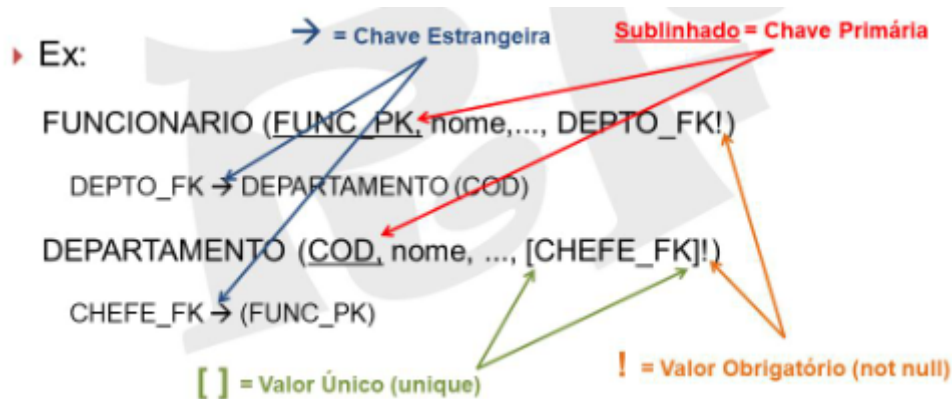
MODELO LÓGICO

- Não é abstrato como o conceitual, tampouco considera aspectos físicos de armazenamento, acesso e desempenho;
- Tem uma sólida base formal (teoria dos conjuntos) e é baseado em conceitos simples (relações com atributos, tuplas e domínios)
 - Relações capturam dados de entidades ou relacionamentos



- Chaves = conceito para identificar/referenciar tuplas
 - Chave candidata = atributo (chave simples) ou concatenação de atributos (chave composta) cujos valores distinguem uma tupla das demais tuplas
 - Deve ser mínima!
 - Chave primária = chave candidata que foi escolhida, usada para selecionar as tuplas de uma relação
 - Não admite valor null
 - Chave estrangeira = atributo ou concatenação de atributos que faz referência a uma chave primária
 - É utilizada para relacionar tuplas de relações
 - Admite valor null (participação opcional)
 - Chave alternativa/secundária = chave candidata que não foi escolhida
 - Não faz relacionamento com chave estrangeira!

- Restrições de integridade = regras sobre os valores armazenados nas relações, têm por objetivo garantir a consistência das relações
 - **Restrições de domínio** = todo valor de um atributo deve ser atômico (simples e mono valorado) e pertencer ao domínio do atributo
 - **Restrições de chave** = todo valor de chave deve ser mínimo e único na relação
 - **Integridade da entidade** = chaves primárias não podem ter valor null
 - **Integridade referencial** = valores de uma chave estrangeira devem aparecer na chave primária da tabela referenciada
- Restrições semânticas = restrições para impor regras de negócio
 - Devem ser implementadas pelos programadores, pois não são garantidas automaticamente
- Notação simplificada:

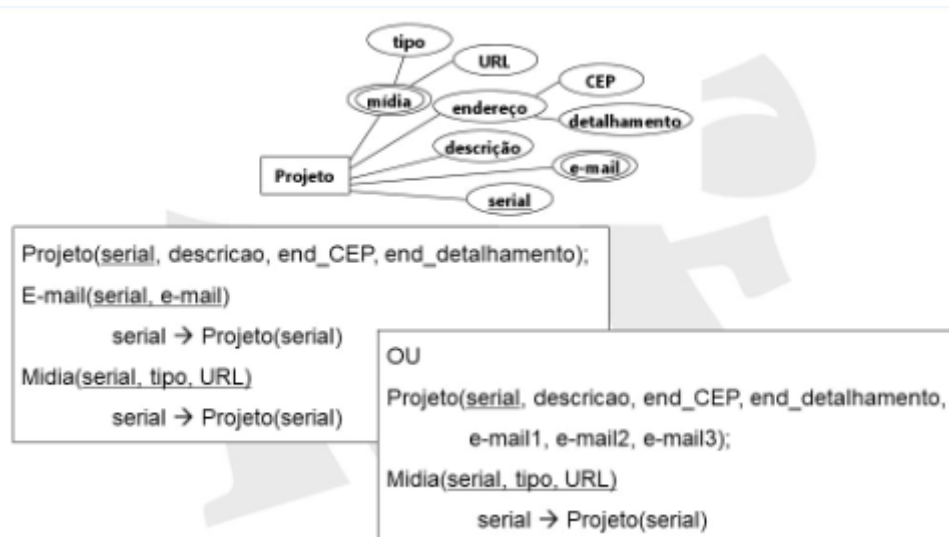


PROJETO RELACIONAL

- Um esquema EER pode gerar N esquemas relacionais
 - Existem várias maneiras de mapear relacionamentos e heranças
- O mapeamento deve priorizar:
 - Evitar junções
 - Diminuir o número de chaves
 - Evitar campos opcionais

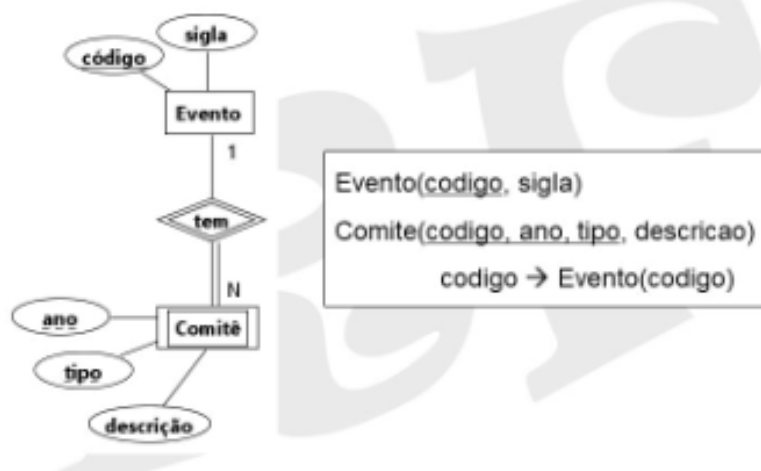
Passos para fazer o mapeamento:

- 1) Mapear as entidades regulares e seus atributos
 - Cada entidade regular é mapeada para uma relação
 - A PK da relação é o atributo identificador da entidade mapeada
 - Cada atributo multivalorado é mapeado para:
 - N atributos (desde que N seja pequeno) ou
 - Uma relação cuja PK é o multivalorado + a PK da relação origem
 - A PK que migrou da relação origem é estrangeira
 - Atributos comum, composto ou derivado são mapeados para atributos da relação



• 2) Mapear as entidades fracas e seus atributos

- Cada entidade fraca é mapeada para uma relação
 - A PK que mapeia a entidade forte migra como FK
 - A PK da relação é formada pela FK + o discriminador (se existir)
- Atributos comum, composto, derivado ou multivalorado seguem os mesmos mapeamentos das entidades regulares



• 3) Mapear as super/subentidades e seus atributos

- Uma relação para cada entidade da herança
 - Pontos fortes e fracos:
 - Funciona bem para qualquer tipo de herança
 - Reduz atributos opcionais
 - Reduz testes para garantir qualidade de dados
 - - Exige junções
 - Cada super/subentidade é mapeada para uma relação
 - A PK de cada relação é o atributo identificador da superentidade mapeada
 - A PK de cada relação que mapeia uma subentidade será FK para a superentidade

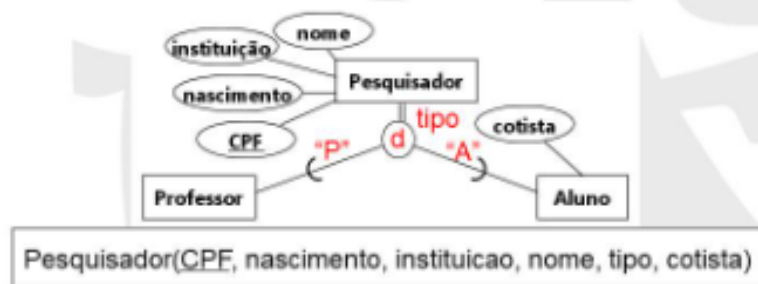
- Atributos comum, composto, derivado ou multivalorado seguem os mesmos mapeamentos das entidades regulares



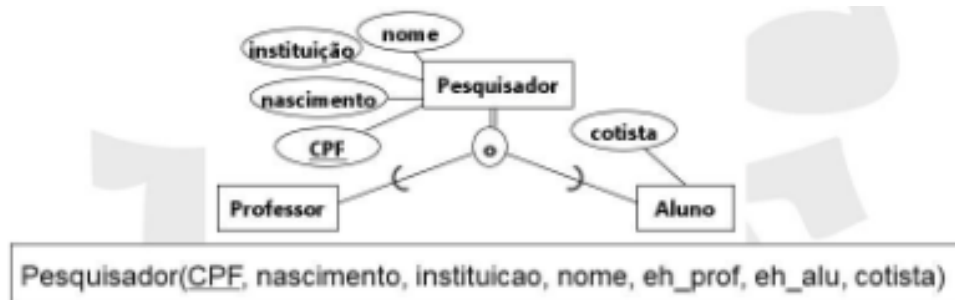
- Uma relação para cada subentidade da herança total
 - Pontos fortes e fracos:
 - - Gera redundância de dados para sobrepostas
 - Reduz atributos opcionais
 - Reduz testes de qualidade
 - Reduz junções



- Uma única relação para toda herança disjunta ou direta
 - Pontos fortes e fracos:
 - - Exige atributos opcionais
 - - Exige testes de qualidade
 - Reduz junções
 - A PK da relação é o identificador da superentidade
 - O predicado da herança se torna um atribuido da relação mapeada
 - Os atributos e relacionamentos das subentidades migram para a relação mapeada
 - Outros atributos são iguais às regulares

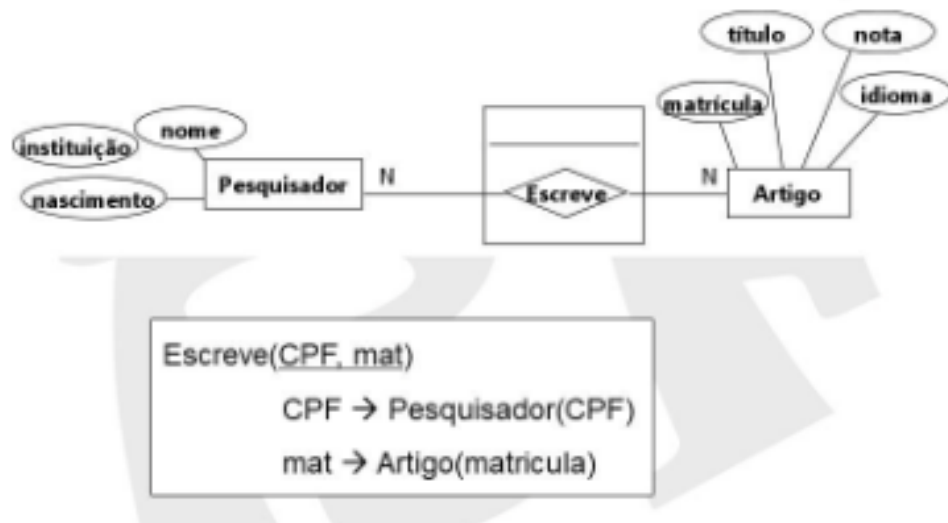


- Uma única relação para toda herança sobreposta
 - Pontos fortes e fracos:
 - - Exige atributos opcionais
 - - Exige testes de qualidade
 - Reduz junções
 - Reduz redundância
 - A PK da relação é o atributo identificador da superentidade mapeada
 - Para cada subentidade, criar na relação mapeada um atributo booleano
 - Os atributos e relacionamentos da superentidade e das subentidades migram para a relação mapeada
 - Outros atributos são iguais às regulares



- **4) Mapear as entidades associativas**

- Cada entidade associativa é mapeada para uma relação
- As PK das relações envolvidas migram com FK obrigatórias
- Os atributos do relacionamento (se houverem) ficam na relação mapeada
- A PK da relação depende do grau e da cardinalidade do relacionamento



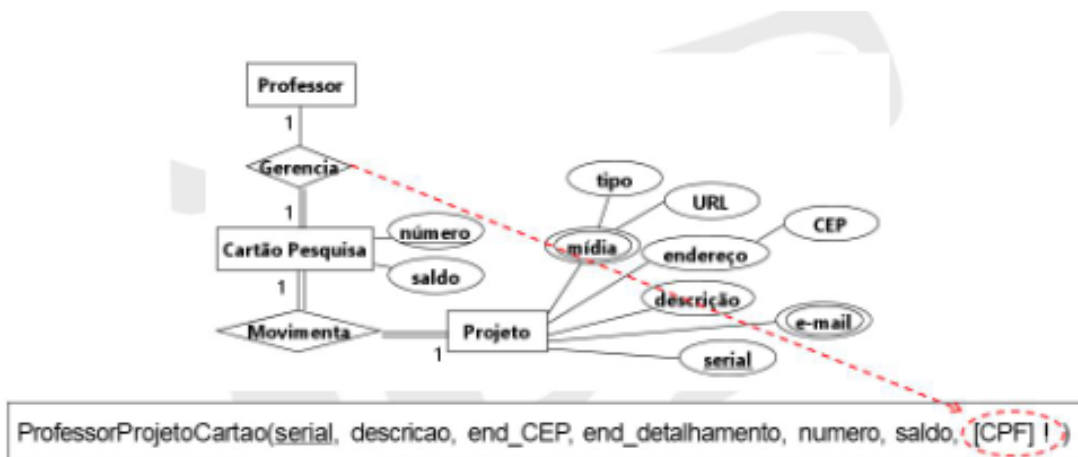
- 5) Mapear os relacionamentos e seus atributos

- Fusão de relações

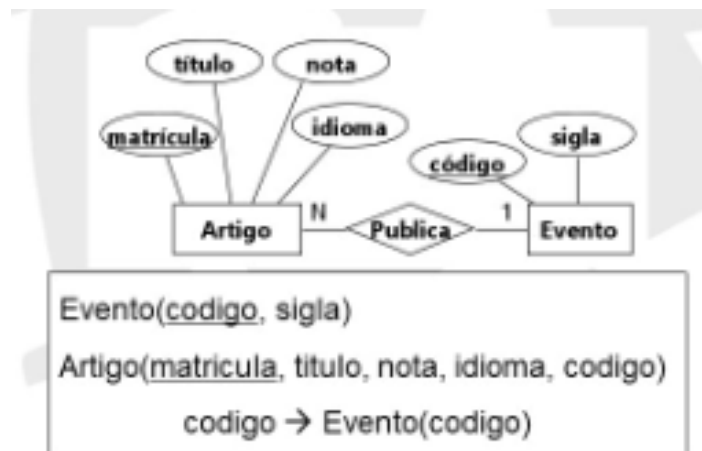
- - Exigem atributos opcionais
- - Exigem testes de qualidade
- Reduz junções
- É melhor no caso 1:1 total-total
 - Fundir as relações em uma única relação
 - A PK da relação fundida deve ser uma das originais
 - Usar []! para a outra PK como AK
 - Para 1:1 total-total:



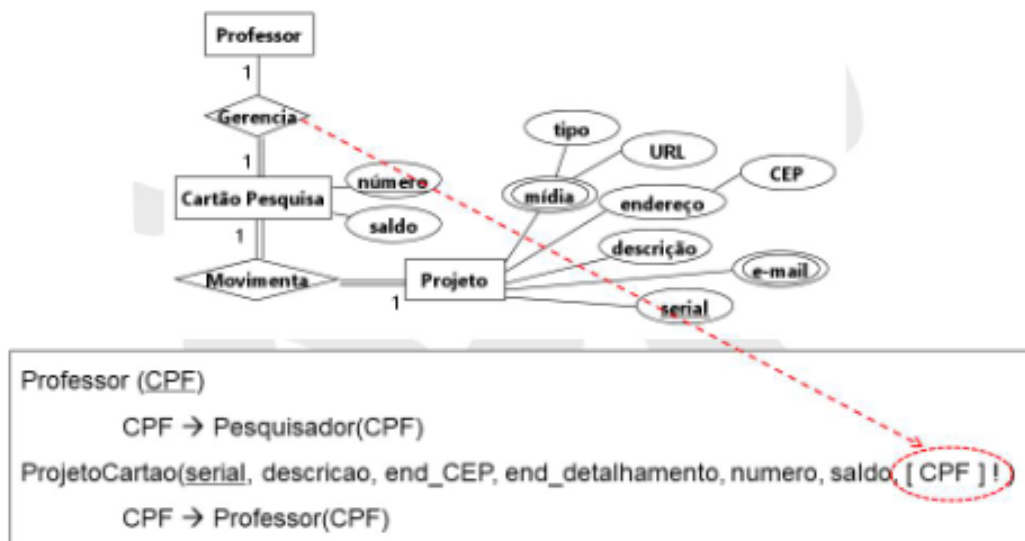
- Caso 1:1 total-parcial:
 - Fundir as relações em uma única relação
 - A PK da relação fundida deve ser a PK da relação original que tem participação parcial
 - Usar [] para definir a outra PK como AK
 - Avaliar custo x benefício



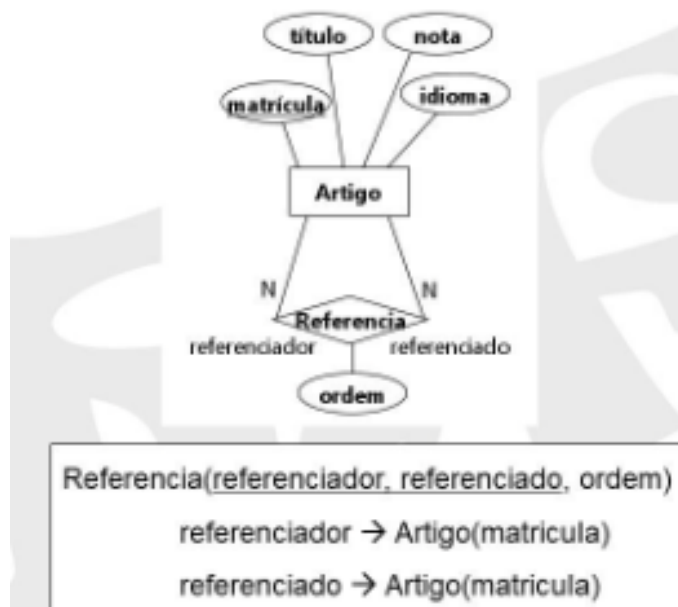
- Adição de chave estrangeira
 - Reduz atributos opcionais
 - Reduz testes de qualidade
 - - Exige junção
 - Melhor caso é 1:N
 - A PK da relação do lado 1 migra como FK para a outra relação
 - Caso o lado N seja total, usar ! para definir a FK como obrigatória
 - Os atributos do relacionamento (se houverem) migram com a PK



- Caso 1:1:
 - Parcial-parcial: a PK de qualquer uma das relações migra como FK única e opcional (usar [] para definir a FK como única)
 - Total-parcial: a PK da relação do lado parcial migra como FK única e obrigatória (usar [] ! para definir a FK como única e obrigatória)
 - Total-total: a PK de qualquer uma das relações migra como FK única e obrigatória (usar [] ! para a FK)
 - Os atributos do relacionamento (se houverem) migram com a PK

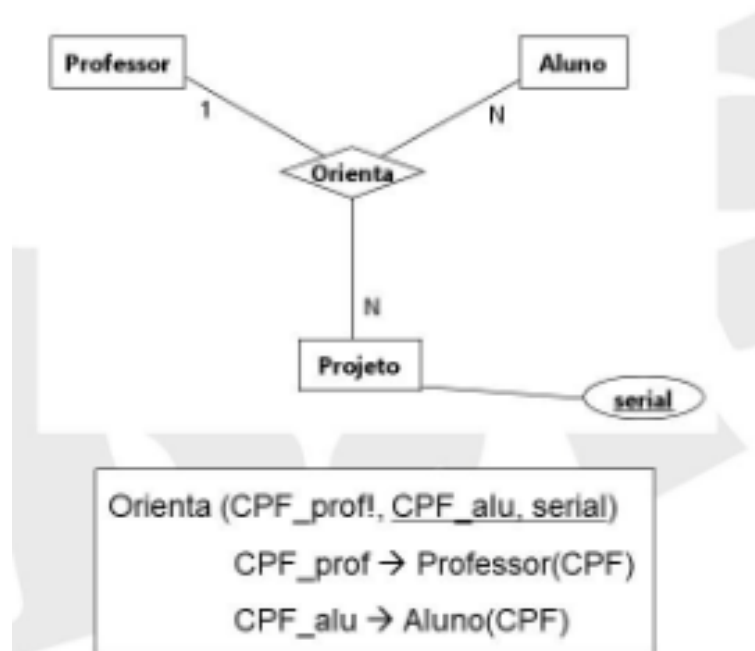


- Criação de relação
 - Reduz atributos opcionais
 - Reduz testes de qualidade
 - - Exige junções
 - Melhor caso M:N:
 - Cada relacionamento M:N é mapeado para uma relação
 - As PK das relações envolvidas migram com FK
 - A composição das FK forma a PK da relação
 - Os atributos do relacionamento (se houverem) ficam na relação mapeada

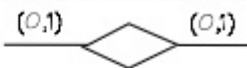











- Caso 1:N (evitar):
 - O relacionamento 1:N é mapeado para uma relação
 - As PK das relações envolvidas migram como FK
 - A FK do lado N torna-se PK

- A outra FK torna-se obrigatória (usar ! para FK)
- Os atributos do relacionamento (se houverem) ficam na relação mapeada
- Caso 1:1 (evitar):
 - O relacionamento 1:1 é mapeado para uma relação
 - As PK das relações envolvidas migram como FK
 - A PK é definida a partir das participações do relacionamento
 - Parcial-parcial = a PK é qualquer uma das FK
 - Total-parcial = a PK é a FK do lado parcial
 - Total-total = a PK é qualquer uma das FK
 - A outra FK se torna AK (usar [] !)
 - Os atributos do relacionamento (se houverem) ficam na relação mapeada
- Relacionamentos N-ários:
 - Cada relacionamento n-ário é mapeado para uma relação
 - As PK das relações envolvidas migram com FK obrigatórias (usar ! [])
 - Os atributos do relacionamento (se houverem) ficam na relação mapeada
 - A PK da relação depende da cardinalidade do relacionamento:
 - N:N:N - PK formada por todas as FK
 - 1:N:N - PK dupla formada pelas FK do lado N
 - 1:1:N e 1:1:1 - Casp raro e complexo



- Slide 42 pra frente = exemplos práticos

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:1			
	±	✓	×
	×	±	✓
	×	±	✓
Relacionamentos 1:n			
	±	✓	×
	±	✓	×
	×	✓	×
	×	✓	×
Relacionamentos n:n			
	✓	×	×
	✓	×	×
	✓	×	×

✓ Alternativa preferida ± Pode ser usada × Não usar