

Ao projetar um esquema de banco de dados, a escolha dos nomes para tipos de entidade, atributos, tipos de relacionamentos e (particularmente) funções nem sempre é simples. É preciso escolher nomes que transmitam, tanto quanto possível, os significados conectados às diferentes construções no esquema. Escolhemos usar *nomes no singular* para os tipos de entidade, em vez de nomes do plural, porque o nome se aplica a cada entidade individual pertencente a esse tipo de entidade. Em nossos diagramas ER, usaremos a convenção de que os nomes do tipo de entidade e tipo de relacionamento são escritos em letras maiúsculas, os nomes de atributo têm apenas a letra inicial em maiúscula e os nomes de papel são escritos em letras minúsculas; Usamos essa convenção na Figura 15.

Como uma prática geral, dada uma descrição narrativa dos requisitos do banco de dados, os *nomes* que aparecem na narrativa tendem a gerar nomes de tipos de entidade, e os *verbos* tendem a indicar nomes de tipos de relacionamento. Os nomes de atributos costumam surgir de nomes adicionais que descrevem os nomes correspondentes para tipos de entidades.

Outra consideração de nomeação envolve a escolha de nomes de relacionamento binário para tornar o diagrama ER do esquema legível da esquerda para a direita e de cima para baixo. Seguimos essa orientação de modo geral na Figura 15. Observe que essa escolha surge porque cada relacionamento binário pode ser descrito começando de qualquer um dos dois tipos de entidades participantes.

## **2. Projeto de banco de dados relacional por mapeamento ER para relacional**

Esta seção discute como **projetar um esquema de banco de dados relacional** com base em um projeto de esquema conceitual. A Figura 14 apresentou uma visão de alto nível do processo de projeto de banco de dados e, nesta seção focamos a etapa de projeto lógico de banco de dados lógico ou mapeamento de modelo de dados do projeto de banco de dados. Apresentaremos os procedimentos para criar um esquema relacional com base em um esquema Entidade-Relacionamento (ER). Nossa discussão relaciona as construções do modelo ER, apresentadas na seção 1, às construções do modelo relacional, apresentadas no Capítulo 1. Muitas ferramentas de engenharia de software auxiliada por computador (CASE) são baseadas no modelo ER, ou outros modelos semelhantes. Muitas ferramentas utilizam diagramas ER ou variações para desenvolver um esquema

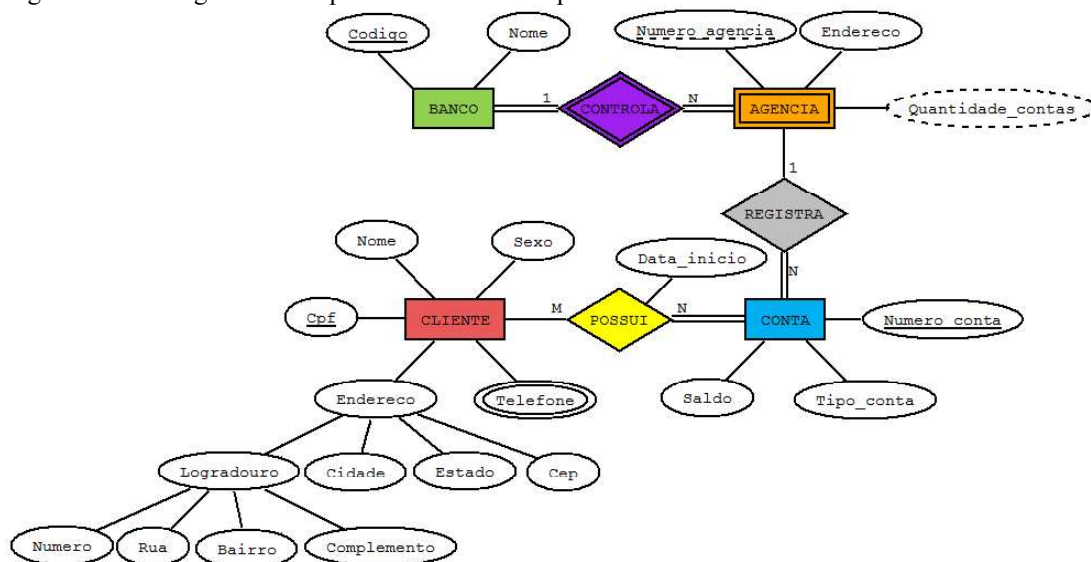
graficamente, e depois o convertem de maneira automática em um esquema de banco de dados relacional na DDL de um SGBD específico, empregando algoritmos semelhantes aos apresentados nesta seção.

Esboçamos um algoritmo de sete etapas na Seção 2.1 para converter as construções básicas do modelo ER – tipos de entidade (forte e fraca), relacionamentos binários (com várias restrições estruturais), relacionamentos  $n$ -ários e atributos (simples, compostos e multivalorados) – em relações.

## 2.1 Algoritmo de mapeamento ER para relacional

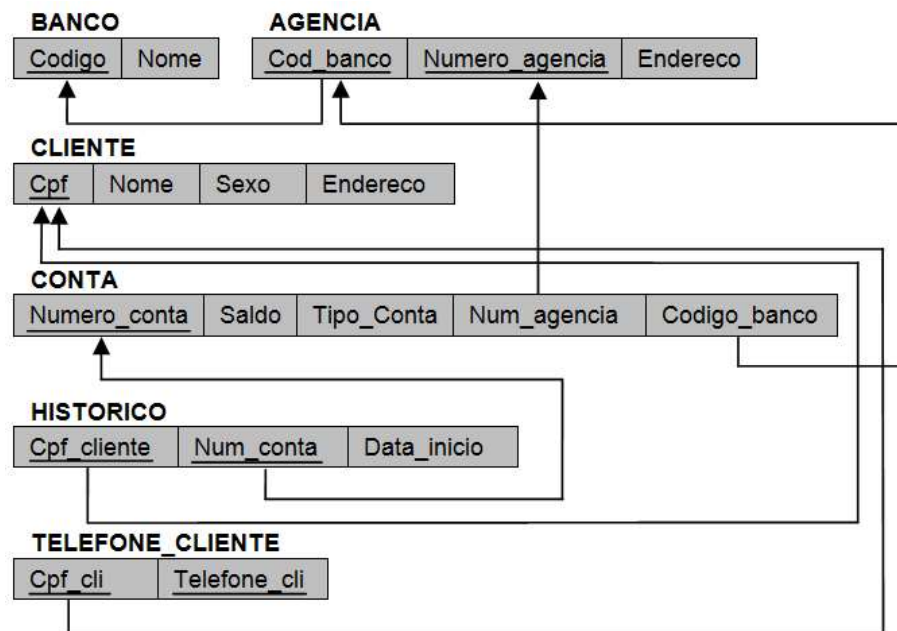
Nesta seção, vamos descrever as etapas de um algoritmo para mapeamento ER para relacional. Usamos o exemplo de banco de dados BANCO para ilustrar o procedimento de mapeamento. O esquema ER BANCO aparece novamente na Figura 32, e o esquema de banco de dados relacional BANCO correspondente aparece na Figura 33 para ilustrar as etapas do mapeamento. Assumimos que o mapeamento criará tabelas com atributos simples de único valor. As restrições do modelo relacional definidas no Capítulo 1, que incluem chaves primárias, chaves únicas (se houver) e restrições de integridade referencial sobre as relações, também serão especificadas nos resultados do mapeamento.

Figura 32 – O diagrama do esquema conceitual ER para o banco de dados BANCO



Fonte: o autor

Figura 33 – Resultado do mapeamento do esquema ER BANCO para um esquema de banco de dados relacional



Fonte: o autor

### Etapa 1: Mapeamento de tipos de entidade regular

Para cada tipo entidade regular (forte)  $E$  no esquema ER, crie uma relação  $R$  que inclua todos os atributos simples de  $E$ . Inclua apenas os atributos de componentes simples de um atributo composto. Escolha um dos atributos-chave de  $E$  como chave primária para  $R$ . Se a chave escolhida de  $E$  for composta, então o conjunto de atributos simples que a compõem juntos formarão a chave primária de  $R$ .

Se várias chaves fossem identificadas para  $E$  durante o projeto conceitual, a informação que descreve os atributos que formam cada chave adicional é mantida a fim de especificar chaves secundárias (únicas) da relação  $R$ . O conhecimento sobre as chaves também é mantido para fins de indexação e outros tipos de análises.

Em nosso exemplo, criamos as relações BANCO, CLIENTE e CONTA na Figura 33 que correspondem aos tipos entidades regulares BANCO, CLIENTE e CONTA da Figura 32. Os atributos de chave estrangeira e relacionamento, se houver, ainda não estão incluídos; eles serão acrescentados durante as etapas seguintes. Estes incluem o atributo Num\_agencia de CONTA. Em nosso exemplo, escolhemos Codigo, Cpf e Numero\_conta como chaves primárias para as relações BANCO, CLIENTE e CONTA, respectivamente.

As relações que são criadas com base no mapeamento dos tipos de entidade às vezes são chamadas **relações de entidade**, pois cada tupla representa uma

instância de uma entidade. O resultado após essa etapa de mapeamento aparece na Figura 34(a).

Figura 34. Ilustração de algumas etapas de mapeamento. (a) Relações de entidade após a etapa 1. (b) Relação de entidade fraca após a etapa 2. (c) Relação de relacionamento após a etapa 5. (d) Relação representando atributo multivalorado após a etapa 6.

(a) **BANCO**

<u>Codigo</u>	Nome
---------------	------

**CLIENTE**

<u>Cpf</u>	Nome	Sexo	Endereco
------------	------	------	----------

**CONTA**

<u>Numero_conta</u>	Saldo	Tipo_Conta
---------------------	-------	------------

(b) **AGENCIA**

<u>Cod_banco</u>	<u>Numero_agencia</u>	Endereco
------------------	-----------------------	----------

(c) **HISTORICO**

<u>Cpf_cliente</u>	<u>Num_conta</u>	Data_inicio
--------------------	------------------	-------------

(d) **TELEFONE\_CLIENTE**

<u>Cpf_cli</u>	<u>Telefone_cli</u>
----------------	---------------------

Fonte: o autor

## Etapa 2: Mapeamento de tipos de entidade fraca

Para cada tipo de entidade fraca  $F$  no esquema ER com tipo de entidade proprietária  $E$ , crie uma relação  $R$  e inclua todos os atributos simples (ou componentes simples dos atributos compostos) de  $F$  como atributos de  $R$ . Além disso, inclua como atributos de chave estrangeira de  $R$  os atributos de chaves primária da(s) relação(ões) que corresponde(m) aos tipos de entidade proprietária. Isso consegue mapear o tipo de relacionamento de identificação de  $F$ . A chave primária de  $R$  é a combinação das chaves primárias dos proprietários e a chave parcial do tipo de entidade fraca  $F$ , se houver.

Se houver um tipo de entidade fraca  $E_2$ , cujo proprietário também é um tipo de entidade  $E_1$ , então  $E_1$  deve ser mapeado antes de  $E_2$  para determinar primeiro a sua chave primária.

Em nosso exemplo, criamos a relação AGENCIA nesta etapa para corresponder ao tipo de entidade fraca AGENCIA (ver Figura 34(b)). Incluímos a chave primária Codigo da relação BANCO – que corresponde ao tipo de entidade proprietária – como um atributo de chave estrangeira de AGENCIA; renomeamos para Cod\_banco, embora

isso não seja necessário. A chave primária da relação AGENCIA é a combinação {Cod\_banco, Numero\_agencia}, pois Numero\_agencia é chave parcial de AGENCIA.

### **Etapa 3: Mapeamento dos tipos de relacionamento binário 1:1**

Para cada tipo de relacionamento binário 1:1  $R$  no esquema ER, identifique as relações  $S$  e  $T$  que correspondem aos tipos de entidade participantes em  $R$ . Escolha uma das relações – digamos,  $S$  – e inclua como chave estrangeira em  $S$  a chave primária de  $T$ . É melhor escolher um tipos de entidade com *participação total* em  $R$  no papel de  $S$ . Inclua todos os atributos simples (ou componentes simples dos atributos compostos) do tipo de relacionamento 1:1  $R$  como atributos de  $S$ .

Em nosso exemplo, não há nenhum tipo de relacionamento binário 1:1.

### **Etapa 4: Mapeamento de tipos de relacionamento binário 1:N**

Para cada tipo relacionamento  $R$  binário regular 1:N, identifique a relação  $S$  que representa o tipo de entidade participante no lado  $N$  do tipo de relacionamento. Inclua como chave estrangeira em  $S$  a chave primária da relação  $T$  que representa o outro tipo de entidade participante em  $R$ ; fazemos isso porque cada instância de entidade no lado  $N$  está relacionada a, no máximo, uma instância de entidade no lado 1 do tipo de relacionamento. Inclua quaisquer atributos simples (ou componentes simples dos atributos compostos) do tipo de relacionamento 1:N como atributo de  $S$ .

Em nosso exemplo, agora mapeamos os tipos de relacionamento 1:N CONTROLA e REGISTRA da Figura 32. Para CONTROLA incluímos a chave primária Codigo da relação BANCO como chave estrangeira na relação AGENCIA e a chamamos de Cod\_banco. Para REGISTRA, incluímos a chave primária – que é composta – da relação AGENCIA como chave estrangeira na relação CONTA e a chamamos de Num\_agencia e Codigo\_banco. Essas chaves estrangeiras são mostradas na Figura 33.

### **Etapa 5: Mapeamento de tipos de relacionamento binário M:N**

Para cada tipo relacionamento  $R$  binário M:N, crie uma nova relação  $S$  para representar  $R$ . Inclua como atributos de chave estrangeira em  $S$  as chaves primárias das relações que representam os tipos de entidade participantes; sua *combinação* formará a chave primária de  $S$ . Inclua também quaisquer atributos simples do tipo de relacionamento M:N (ou componentes simples dos atributos compostos) como

atributos de  $S$ . Observe que não podemos representar um tipo de relacionamento M:N por um único atributo de chave estrangeira em uma das relações participantes (como fizemos para os tipos de relacionamentos 1:1 ou 1:N) devido à razão de cardinalidade M:N; temos de criar uma *relação de relacionamento*  $S$  separada.

Em nosso exemplo, mapeamos o tipo de relacionamento M:N POSSUI da Figura 32 criando a relação HISTORICO na Figura 33. Incluímos as chaves primárias das relações CLIENTE e CONTA como chaves estrangeiras em HISTORICO e as renomeamos como Cpf\_cliente e Num\_conta, respectivamente. Também incluímos um atributo Data\_inicio em HISTORICO para representar o atributo Data\_inicio do tipo de relacionamento. A chave primária da relação HISTORICO é a combinação dos atributos de chave estrangeiras {Cpf\_cliente, Num\_conta}. Essa **relação de relacionamento** aparece na Figura 34(c).

A opção de propagação (CASCADE) para a ação de disparo referencial (ver Capítulo 1) deve ser especificada sobre as chaves estrangeiras na relação correspondente ao relacionamento  $R$ , pois cada instância de relacionamento tem uma dependência de existência sobre cada uma das entidades a que ela se relaciona. Isso pode ser usado tanto para ON UPDATE quanto para ON DELETE.

Observe que sempre podemos mapear relacionamentos 1:1 ou 1:N de uma maneira semelhante aos relacionamentos M:N usando a técnica de referência cruzada (relação de relacionamento), conforme discutimos anteriormente. Essa alternativa é particularmente útil quando existem poucas instâncias de relacionamentos, a fim de evitar valores NULL em chaves estrangeiras. Nesse caso, a chave primária da relação de relacionamento será *apenas uma* das chaves estrangeiras que referenciam as relações da entidade participante. Para um relacionamento 1:N, a chave primária da relação de relacionamento será a chave estrangeira que referencia a relação de entidade no lado N. Para um relacionamento 1:1, qualquer chave estrangeira pode ser usada como chave primária da relação de relacionamento.

## **Etapas 6: Mapeamento de atributos multivalorados**

Para cada atributo multivalorado  $A$ , crie uma relação  $R$ . Essa relação  $R$  incluirá um atributo correspondente a  $A$ , mais o atributo da chave primária  $Ch$  – como uma chave estrangeira em  $R$  – da relação que representa o tipo de entidade ou tipo de relacionamento que tem  $A$  como atributo multivalorado. A chave primária de

$R$  é a combinação de  $A$  e  $Ch$ . Se o atributo multivalorado for composto, incluímos seus componentes simples.

Em nosso exemplo, criamos uma relação TELEFONE\_CLIENTE (ver Figura 34(d)). O atributo Telefone\_cli representa o atributo multivalorado Telefone de CLIENTE, enquanto Cpf\_Cli – como chave estrangeira — representa a chave primária da relação CLIENTE. A chave primária de TELEFONE\_CLIENTE é a combinação de {Telefone\_cli, Cpf\_cli}. Uma tupla separada existirá em TELEFONE\_CLIENTE para cada telefone de um cliente.

Também devemos observar que a chave de  $R$ , ao mapear um atributo composto, multivalorado, requer alguma análise do significado dos atributos componentes. Em alguns casos, quando um atributo multivalorado é composto, somente alguns dos atributos componentes são exigidos para fazer parte da chave de  $R$ . Esses atributos são semelhantes à chave parcial de um tipo de entidade fraca que corresponde ao atributo multivalorado (ver Seção 1.5).

A Figura 33 mostra o esquema de banco de dados relacional BANCO obtido com as etapas 1 a 6. Observe que ainda não discutimos o mapeamento de tipos de relacionamentos  $n$ -ário ( $n > 2$ ), pois ele não existe na Figura 32. Estes são mapeados de um modo semelhante aos tipos de relacionamentos  $M:N$ , incluindo a etapa adicional a seguir no algoritmo de mapeamento.

### **Etapla 7: Mapeamento de tipos de relacionamento $n$ -ário**

Para cada tipo de relacionamento  $n$ -ário  $R$ , onde  $n > 2$ , crie uma relação  $S$  para representar  $R$ . Inclua como atributos de chave estrangeira em  $S$  as chaves primárias das relações que representam os tipos de entidade participantes. Inclua também quaisquer atributos simples do tipo de relacionamento  $n$ -ário (ou componentes simples de atributos compostos) como atributo de  $S$ . A chave primária de  $S$  normalmente é uma combinação de todas as chaves estrangeiras que referenciam as relações representando os tipos de entidade participantes. Porém, se as restrições de cardinalidade sobre qualquer um dos tipos de entidade  $E$  participantes em  $R$  for 1, então a chave primária de  $S$  não deve incluir o atributo de chave estrangeira que referencia a relação  $E'$  correspondente a  $E$ .

## 2.2 Discussão e resumo do mapeamento para construções no modelo ER

A Tabela 2 resume as correspondências entre as construções e restrições do modelo ER e relacional.

Tabela 2 – Correspondência entre os modelos ER e Relacional

MODELO ER	MODELO RELACIONAL
Tipo de entidade	Relação de <i>entidade</i>
Tipo de relacionamento 1:1 ou 1:N	Chave estrangeira (ou relação de <i>relacionamento</i> )
Tipo de relacionamento M:N	Relação de <i>relacionamento</i> e <i>duas</i> chaves estrangeiras
Tipo de relacionamento <i>n</i> -ário	Relação de <i>relacionamento</i> e <i>n</i> chaves estrangeiras
Atributo simples	Atributo
Atributo composto	Conjunto de atributos Componentes simples
Atributo multivalorado	Relação e chave estrangeira
Conjunto de valores	Domínio
Atributo-chave	Chave primária (ou secundária)

Fonte: o autor

Um dos principais pontos a observar em um esquema relacional, ao contrário de um esquema ER, é que os tipos de relacionamento não são representados explicitamente. Em vez disso, eles são representados por dois atributos *A* e *B*, um é uma chave primária e o outro é uma chave estrangeira (no mesmo domínio) incluída em duas relações *S* e *T*. Duas tuplas em *S* e *T* são relacionadas quando têm o mesmo valor para *A* e *B*. Pelo uso da operação EQUIJOIN (ou NATURAL JOIN — junção natural —, se os dois atributos de junção possuírem o mesmo nome) em *S.A* e *T.B*, podemos combinar todos os pares de tuplas relacionados entre *S* e *T* e, assim, efetivar o relacionamento. Quando um tipo relacionamento 1:1 ou 1:N está envolvido, normalmente uma única operação de junção é suficiente. Para um tipo relacionamento binário N:M, duas operações de junção são necessárias, ao passo que, nos tipos relacionamentos *n*-ários, *n* junções são essenciais para materializar completamente as instâncias do relacionamento.

Por exemplo, para formar uma relação que inclui o nome do cliente, número da conta e data de início que o cliente tornou-se titular da conta, precisamos conectar cada tupla CLIENTE às tuplas CONTA relacionadas por meio da relação HISTORICO na Figura 33. Logo, precisamos aplicar a operação EQUIJUNÇÃO às relações CLIENTE e HISTORICO com a condição de junção *Cpf = Cpf\_cliente*, e depois aplicar outra



operação EQUIJUNÇÃO à relação resultante e a relação CONTA com a condição de junção `Numero_conta = Num_conta`. Em geral, quando vários relacionamentos precisam ser examinados, diversas operações de junção precisam ser especificadas. Um usuário de banco de dados relacional sempre precisa estar ciente dos atributos de chave estrangeira para poder usá-los corretamente na combinação de tuplas relacionadas de duas ou mais relações. Isso às vezes é considerado uma desvantagem do modelo relacional, porque as correspondências de chave estrangeira/chave primária nem sempre são óbvias pela inspeção dos esquemas relacionais. Se uma EQUIJUNÇÃO for realizada entre atributos de duas relações que não representam um relacionamento de chave estrangeira/chave primária, o resultado pode com frequência ser sem sentido e levar a dados falsos (espúrios).

No esquema relacional, criamos uma relação separada para *cada* atributo multivalorado. Para uma entidade em particular com um conjunto de valores para o atributo multivalorado, o valor do atributo-chave da entidade é repetido uma vez para cada valor do atributo multivalorado em uma tupla separada, pois o modelo relacional básico *não* permite valores múltiplos (uma lista, ou um conjunto de valores) para um atributo em uma única tupla. Por exemplo, um cliente que tem três telefones, existem três tuplas na relação TELEFONE\_CLIENTE; cada tupla especifica um dos telefones. Em nosso exemplo, aplicamos EQUIJUNÇÃO a TELEFONE\_CLIENTE e CLIENTE nos atributos Cpf\_cli e Cpf para obter os valores de todos os telefones junto com outros atributos de CLIENTE. Na relação resultante, os valores dos outros atributos de CLIENTE são repetidos em tuplas separadas para cada telefone de um cliente.