

3.2 Restrições em modelo relacional e esquemas de bancos de dados relacionais

Até aqui, discutimos as características de relações isoladas. No banco de dados relacional, normalmente haverá muitas relações, e as tuplas nessas relações costumam estar relacionadas de várias maneiras. O estado do banco de dados inteiro corresponderá aos estados de todas as suas relações em determinado ponto no tempo. Em geral, existem muitas **restrições** (ou *constraints*) sobre os valores reais em um estado do banco de dados. Essas restrições são derivadas das regras no minimundo que o banco de dados representa, conforme discutimos na Seção 1.6.8.

Nesta seção, discutiremos as diversas restrições sobre os dados que podem ser especificadas em um banco de dados relacional na forma de restrições. As restrições nos bancos de dados geralmente podem ser divididas em três categorias principais:

1. Restrições que são inerentes no modelo de dados. Chamamos estas de **restrições inerentes baseadas no modelo** ou **restrições implícitas**.
2. Restrições que podem ser expressas diretamente nos esquemas do modelo de dados, em geral especificando-as na DDL (linguagem de definição de dados; ver Seção 2.3.1). Chamamos estas de **restrições baseadas em esquema** ou **restrições explícitas**.
3. Restrições que *não podem* ser expressas diretamente nos esquemas do modelo de dados, e, portanto, devem ser expressas e impostas pelos programas de aplicação. Chamamos estas de **restrições baseadas na aplicação** ou **restrições semânticas** ou **regras de negócios**.

As características das relações que discutimos na Seção 3.1.2 são as restrições inerentes do modelo relacional e pertencem à primeira categoria. Por exemplo, a restrição de que uma relação não pode ter tuplas duplicadas é uma restrição inerente. As restrições que discutimos nesta seção são da segunda categoria, a saber, restrições que podem ser expressas no esquema do modelo relacional por meio da DDL. As restrições da terceira categoria são mais gerais, relacionam-se ao significado e também ao comportamento dos atributos, e são difíceis de expressar e impor dentro do modelo de dados, de modo que normalmente são verificadas nos programas de aplicação que realizam as atualizações no banco de dados.

Outra categoria importante de restrições é a de *dependências de dados*, que incluem *dependências funcionais* e *dependências multivaloradas*. Elas são usadas principalmente para testar a 'virtude' do pro-

jeto de um banco de dados relacional e em um processo chamado *normalização*, que será discutido nos capítulos 15 e 16.

As restrições baseadas em esquema incluem restrições de domínio, restrições de chave, restrições sobre NULLs, restrições de integridade de entidade e restrições de integridade referencial.

3.2.1 Restrições de domínio

As restrições de domínio especificam que, dentro de cada tupla, o valor de cada atributo A deve ser um valor indivisível do domínio $\text{dom}(A)$. Já discutimos as maneiras como os domínios podem ser especificados na Seção 3.1.1. Os tipos de dados associados aos domínios normalmente incluem os tipos de dados numéricos padrão para inteiros (como short integer, integer e long integer) e números reais (float e double). Caracteres, booleanos, cadeia de caracteres de tamanho fixo e cadeia de caracteres de tamanho variável também estão disponíveis, assim como data, hora, marcador de tempo, moeda ou outros tipos de dados especiais. Outros domínios possíveis podem ser descritos por um subintervalo dos valores de um tipo de dados ou como um tipo de dado enumerado, em que todos os valores possíveis são explicitamente listados. Em vez de descrevê-los com detalhes aqui, discutiremos os tipos de dados oferecidos pelo padrão relacional SQL na Seção 4.1.

3.2.2 Restrições de chave e restrições sobre valores NULL

No modelo relacional formal, uma *relação* é definida como um *conjunto de tuplas*. Por definição, todos os elementos de um conjunto são distintos; logo, todas as tuplas em uma relação também precisam ser distintas. Isso significa que duas tuplas não podem ter a mesma combinação de valores para *todos* os seus atributos. Normalmente, existem outros **subconjuntos de atributos** de um esquema de relação R com a propriedade de que duas tuplas em qualquer estado de relação r de R não deverão ter a mesma combinação de valores para esses atributos. Suponha que indiquemos um subconjunto de atributos desse tipo como SCh ; então, para duas tuplas *distintas* quaisquer t_1 e t_2 em um estado de relação r de R , temos a restrição de que:

$$t_1[SCh] \neq t_2[SCh]$$

Qualquer conjunto de atributos SCh desse tipo é chamado de **superchave** do esquema de relação R . Uma superchave SCh especifica uma *restrição de exclusividade* de que duas tuplas distintas em qualquer estado r de R não podem ter o mesmo valor de SCh . Cada relação tem pelo menos uma superchave padrão — o

conjunto de todos os seus atributos. Contudo, uma superchave pode ter atributos redundantes, de modo que um conceito mais útil é o de uma *chave*, que não tem redundância. Uma **chave** *Ch* de um esquema de relação *R* é uma superchave de *R* com a propriedade adicional de que a remoção de qualquer atributo *A* de *Ch* deixa um conjunto de atributos *Ch'* que não é mais uma superchave de *R*. Logo, uma chave satisfaz duas propriedades:

1. Duas tuplas distintas em qualquer estado da relação não podem ter valores idênticos para (todos) os atributos na chave. Essa primeira propriedade também se aplica a uma superchave.
2. Ela é uma *superchave mínima* — ou seja, uma superchave da qual não podemos remover nenhum atributo e ainda mantemos uma restrição de exclusividade na condição 1. Essa propriedade não é exigida por uma superchave.

Embora a primeira propriedade se aplique a chaves e superchaves, a segunda propriedade é exigida apenas para chaves. Assim, uma chave também é uma superchave, mas não o contrário. Considere a relação ALUNO da Figura 3.1. O conjunto de atributos {Cpf} é uma chave de ALUNO porque duas tuplas de aluno não podem ter o mesmo valor para Cpf.⁸ Qualquer conjunto de atributos que inclua Cpf — por exemplo, {Cpf, Nome, Idade} — é uma superchave. No entanto, a superchave {Cpf, Nome, Idade} não é uma chave de ALUNO, pois remover Nome ou Idade, ou ambos, do conjunto ainda nos deixa com uma superchave. Em geral, qualquer superchave formada com base em um único atributo também é uma chave. Uma chave com múltiplos atributos precisa exigir que *todos* os seus atributos juntos tenham uma propriedade de exclusividade.

O valor de um atributo de chave pode ser usado para identificar exclusivamente cada tupla na relação. Por exemplo, o valor de Cpf 305.610.243-51 identifica exclusivamente a tupla correspondente a Bruno Braga na relação ALUNO. Observe que um conjunto de atributos constituindo uma chave é uma propriedade do esquema de relação; essa é uma restrição que deve ser mantida sobre *cada* estado de relação válido do esquema. Uma chave é determinada com base no significado dos atributos, e a propriedade é *invariável no tempo*: ela precisa permanecer verdadeira quando inserimos novas tuplas na relação. Por exemplo, não podemos e não deve-

CARRO

Placa	Numero_chassi	Marca	Modelo	Ano
Itatiaia ABC-7039	A6935207586	Volkswagen	Gol	02
Itu TVP-3470	B4369668697	Chevrolet	Corsa	05
Santos MPO-2902	X8355447376	Fiat	Uno	01
Itanhaem TFY-6858	C4374268458	Chevrolet	Celta	99
Itatiba RSK-6279	Y8293586758	Renault	Clio	04
Atibaia RSK-6298	U0283657858	Volkswagen	Parati	04

Figura 3.4

A relação CARRO, com duas chaves candidatas: Placa e Numero_chassi.

mos designar o atributo Nome da relação ALUNO da Figura 3.1 como uma chave porque é possível que dois alunos com nomes idênticos existam em algum ponto em um estado válido.⁹

Em geral, um esquema de relação pode ter mais de uma chave. Nesse caso, cada uma das chaves é chamada de **chave candidata**. Por exemplo, a relação CARRO na Figura 3.4 tem duas chaves candidatas: Placa e Numero_chassi. É comum designar uma das chaves candidatas como **chave primária** da relação. Essa é a chave candidata cujos valores são usados para *identificar* tuplas na relação. Usamos a convenção de que os atributos que formam a chave primária de um esquema de relação são sublinhados, como mostra a Figura 3.4. Observe que, quando um esquema de relação tem várias chaves candidatas, a escolha de uma para se tornar a chave primária é um tanto quanto arbitrária; porém, normalmente é melhor escolher uma chave primária com um único atributo ou um pequeno número de atributos. As outras chaves candidatas são designadas como **chaves únicas** (*unique keys*), e não são sublinhadas.

Outra restrição sobre os atributos especifica se valores NULL são permitidos ou não. Por exemplo, se cada tupla de ALUNO precisar ter um valor válido, diferente de NULL, para o atributo Nome, então Nome de ALUNO é restrito a ser NOT NULL.

3.2.3 Bancos de dados relacionais e esquemas de banco de dados relacional

As definições e restrições que discutimos até aqui se aplicam a relações isoladas e seus atributos. Um

⁸ Observe que Cpf também é uma superchave.

⁹ Os nomes às vezes são usados como chaves, mas, nesse caso, algum artefato — como anexar um número ordinal — precisa ser usado para distinguir esses nomes idênticos.

banco de dados relacional costuma conter muitas relações, com tuplas nas relações que estão relacionadas de várias maneiras. Nesta seção, definimos um banco de dados relacional e um esquema de banco de dados relacional.

Um **esquema de banco de dados relacional** S é um conjunto de esquemas de relação $S = \{R_1, R_2, \dots, R_m\}$ e um conjunto de **restrições de integridade** RI. Um **estado de banco de dados relacional**¹⁰ DB de S é um conjunto de estados de relação $DB = \{r_1, r_2, \dots, r_m\}$, tal que cada r_i é um estado de R_i e tal que os estados da relação r_i satisfazem as restrições de integridade especificadas em RI. A Figura 3.5 mostra um esquema de banco de dados relacional que chamamos de EMPRESA = {FUNCIONARIO, DEPARTAMENTO, LOCALIZACAO_DEP, PROJETO, TRABALHA_EM, DEPENDENTE}. Os atributos sublinhados representam chaves primárias. A Figura 3.6 mostra um estado de banco de dados relacional correspondente ao esquema EMPRESA. Usaremos esse esquema e estado de banco de dados neste capítulo e nos capítulos 4 a 6 para desenvolver consultas de exemplo em diferentes linguagens relacionais.

Quando nos referimos a um banco de dados relacional, implicitamente incluímos seu esquema e seu estado atual. Um estado de banco de dados que não obedece a todas as restrições de integridade é chamado de **estado inválido**, e um estado que satisfaz a todas as restrições no conjunto definido de restrições de integridade RI é chamado de **estado válido**.

Na Figura 3.5, o atributo Dnumero em DEPARTAMENTO e LOCALIZACAO_DEP significa o conceito do mundo real — o número dado a um departamento. O mesmo conceito é chamado Dnr em FUNCIONARIO e Dnum em PROJETO. Os atributos que representam o mesmo conceito do mundo real podem ou não ter nomes idênticos em diferentes relações. Como alternativa, os atributos que representam diferentes conceitos podem ter o mesmo nome em diferentes relações. Por exemplo, poderíamos ter usado o nome de atributo Nome para Projnome de PROJETO e Dnome de DEPARTAMENTO; nesse caso, teríamos dois atributos compartilhando o mesmo nome, mas representando diferentes conceitos do mundo real — nomes de projeto e nomes de departamento.

FUNCIONARIO

Pnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----

DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

LOCALIZACAO_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

PROJETO

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

TRABALHA_EM

<u>Fcpf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

DEPENDENTE

<u>Fcpf</u>	<u>Nome_dependente</u>	Sexo	Datanasc	Parentesco
-------------	------------------------	------	----------	------------

Figura 3.5

Diagrama de esquema para o esquema de banco de dados relacional EMPRESA.

¹⁰ Um **estado** de banco de dados relacional às vezes é chamado de **instância** de banco de dados relacional. No entanto, como mencionamos anteriormente, não usaremos o termo **instância** porque ele também se aplica a tuplas isoladas.

Em algumas versões antigas do modelo relacional, era feita uma suposição de que o mesmo conceito do mundo real, quando representado por um atributo, teria nomes de atributo *idênticos* em todas as relações. Isso cria problemas quando o mesmo conceito do mundo real é usado em diferentes papéis (significados) na mesma relação. Por exemplo, o conceito de Cadastro de Pessoa Física aparece duas vezes na relação FUNCIONARIO da Figura 3.5: uma no papel do CPF do funcionário e outra no papel do CPF do supervisor. Precisamos dar-lhes nomes de atributo distintos — Cpf e Cpf_supervisor, respectivamente — porque eles aparecem na mesma relação e a fim de distinguir seu significado.

Cada SGBD relacional precisa ter uma linguagem de definição de dados (DDL) para estabelecer um esquema de banco de dados relacional. Os SGBDs relacionais atuais costumam usar principalmente SQL para essa finalidade. Apresentaremos a DDL SQL nas seções 4.1 e 4.2.

Restrições de integridade são especificadas em um esquema de banco de dados e espera-se que sejam mantidas em cada estado de banco de dados válido desse esquema. Além das restrições de domínio, chave e NOT NULL, dois outros tipos de restrições são considerados parte do modelo relacional: integridade de entidade e integridade referencial.

3.2.4 Integridade, integridade referencial e chaves estrangeiras

A **restrição de integridade de entidade** afirma que nenhum valor de chave primária pode ser NULL. Isso porque o valor da chave primária é usado para identificar tuplas individuais em uma relação. Ter valores NULL para a chave primária implica que não podemos identificar algumas tuplas. Por exemplo, se duas ou mais tuplas tivessem NULL para suas chaves primárias, não conseguiríamos distingui-las ao tentar referenciá-las por outras relações.

As restrições de chave e as restrições de integridade de entidade são especificadas sobre relações individuais. A **restrição de integridade referencial** é especificada entre duas relações e usada para manter a consistência entre tuplas nas duas relações. Informalmente, a restrição de integridade referencial afirma que uma tupla em uma relação que referencia outra relação precisa se referir a uma *tupla existente* nessa relação. Por exemplo, na Figura 3.6, o atributo Dnr de FUNCIONARIO fornece o número de departamento para o qual cada funcionário trabalha; logo, seu valor em cada tupla FUNCIONARIO precisa combinar com o valor de Dnumero de alguma tupla na relação DEPARTAMENTO.

Para definir a integridade referencial de maneira mais formal, primeiro estabelecemos o conceito de uma *chave estrangeira* (ChE — *foreign key*). As condições para uma chave estrangeira, dadas a seguir, especificam a restrição de integridade referencial entre os dois esquemas de relação R_1 e R_2 . Um conjunto de atributos ChE no esquema de relação R_1 é uma **chave estrangeira** de R_1 que **referencia** a relação R_2 se ela satisfizer as seguintes regras:

1. Os atributos em ChE têm o mesmo domínio (ou domínios) que os atributos de chave primária ChP de R_2 ; diz-se que os atributos ChE **referenciam** ou **referem-se** à relação R_2 .
2. Um valor de ChE em uma tupla t_1 do estado atual $r_1(R_1)$ ocorre como um valor de ChE para alguma tupla t_2 no estado atual $r_2(R_2)$ ou é NULL. No primeiro caso, temos $t_1[\text{ChE}] = t_2[\text{ChP}]$, e dizemos que a tupla t_1 **referencia** ou **refere-se** à tupla t_2 .

Nessa definição, R_1 é chamada de **relação que referencia** e R_2 é a **relação referenciada**. Se essas condições se mantiverem, diz-se que é mantida uma **restrição de integridade referencial** de R_1 para R_2 . Em um banco de dados de muitas relações, normalmente existem muitas restrições de integridade referencial.

Para especificar essas restrições, primeiro devemos ter um conhecimento claro do significado ou papel que cada atributo, ou conjunto de atributos, que fazem parte nos diversos esquemas de relação do banco de dados. As restrições de integridade referencial surgem com frequência dos *relacionamentos entre as entidades* representadas pelos esquemas de relação. Por exemplo, considere o banco de dados mostrado na Figura 3.6. Na relação FUNCIONARIO, o atributo Dnr refere-se ao departamento para o qual um funcionário trabalha; portanto, designamos Dnr para ser a chave estrangeira de FUNCIONARIO que referencia a relação DEPARTAMENTO. Isso significa que um valor de Dnr em qualquer tupla t_1 da relação FUNCIONARIO precisa combinar com um valor da chave primária de DEPARTAMENTO — o atributo Dnumero — em alguma tupla t_2 da relação DEPARTAMENTO, ou o valor de Dnr *pode ser NULL* se o funcionário não pertencer a um departamento ou for atribuído a um departamento mais tarde. Por exemplo, na Figura 3.6, a tupla para o funcionário ‘João Silva’ referencia a tupla para o departamento ‘Pesquisa’, indicando que ‘João Silva’ trabalha para esse departamento.

Observe que uma chave estrangeira pode *se referir a sua própria relação*. Por exemplo, o atributo Cpf_supervisor em FUNCIONARIO refere-se ao supervisor de um funcionário; este é outro funcionário, representado por uma tupla na relação FUNCIONARIO.

FUNCIONARIO

Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Alice	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
Pesquisa	5	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

LOCALIZACAO_DEP

Dnumero	Dlocal
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

TRABALHA_EM

Fcpf	Pnr	Horas
12345678966	1	32,5
12345678966	2	7,5
66688444476	3	40,0
45345345376	1	20,0
45345345376	2	20,0
33344555587	2	10,0
33344555587	3	10,0
33344555587	10	10,0
33344555587	20	10,0
99988777767	30	30,0
99988777767	10	10,0
98798798733	10	35,0
98798798733	30	5,0
98765432168	30	20,0
98765432168	20	15,0
88866555576	20	NULL

PROJETO

Projnome	Projnumero	Projlocal	Dnum
ProdutoX	1	Santo André	5
ProdutoY	2	Itu	5
ProdutoZ	3	São Paulo	5
Informatização	10	Mauá	4
Reorganização	20	São Paulo	1
Novosbenefícios	30	Mauá	4

DEPENDENTE

Fcpf	Nome_dependente	Sexo	Datanasc	Parentesco
33344555587	Alicia	F	05-04-1986	Filha
33344555587	Tiago	M	25-10-1983	Filho
33344555587	Janaína	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha
12345678966	Elizabeth	F	05-05-1967	Esposa

Figura 3.6 Um estado de banco de dados possível para o esquema de banco de dados relacional EMPRESA.

Logo, Cpf_supervisor é uma chave estrangeira que referencia a própria relação FUNCIONARIO. Na Figura 3.6, a tupla que o funcionário ‘João Silva’ referencia é a tupla do funcionário ‘Fernando Wong’, indicando que ‘Fernando Wong’ é o supervisor de ‘João Silva’.

Podemos *exibir em forma de diagrama as restrições de integridade referencial*, desenhando um arco direcionado de cada chave estrangeira para a relação que ela referencia. Para ficar mais claro, a ponta da seta pode apontar para a chave primária da relação referenciada. A Figura 3.7 mostra o esquema da Fi-

FUNCIONARIO

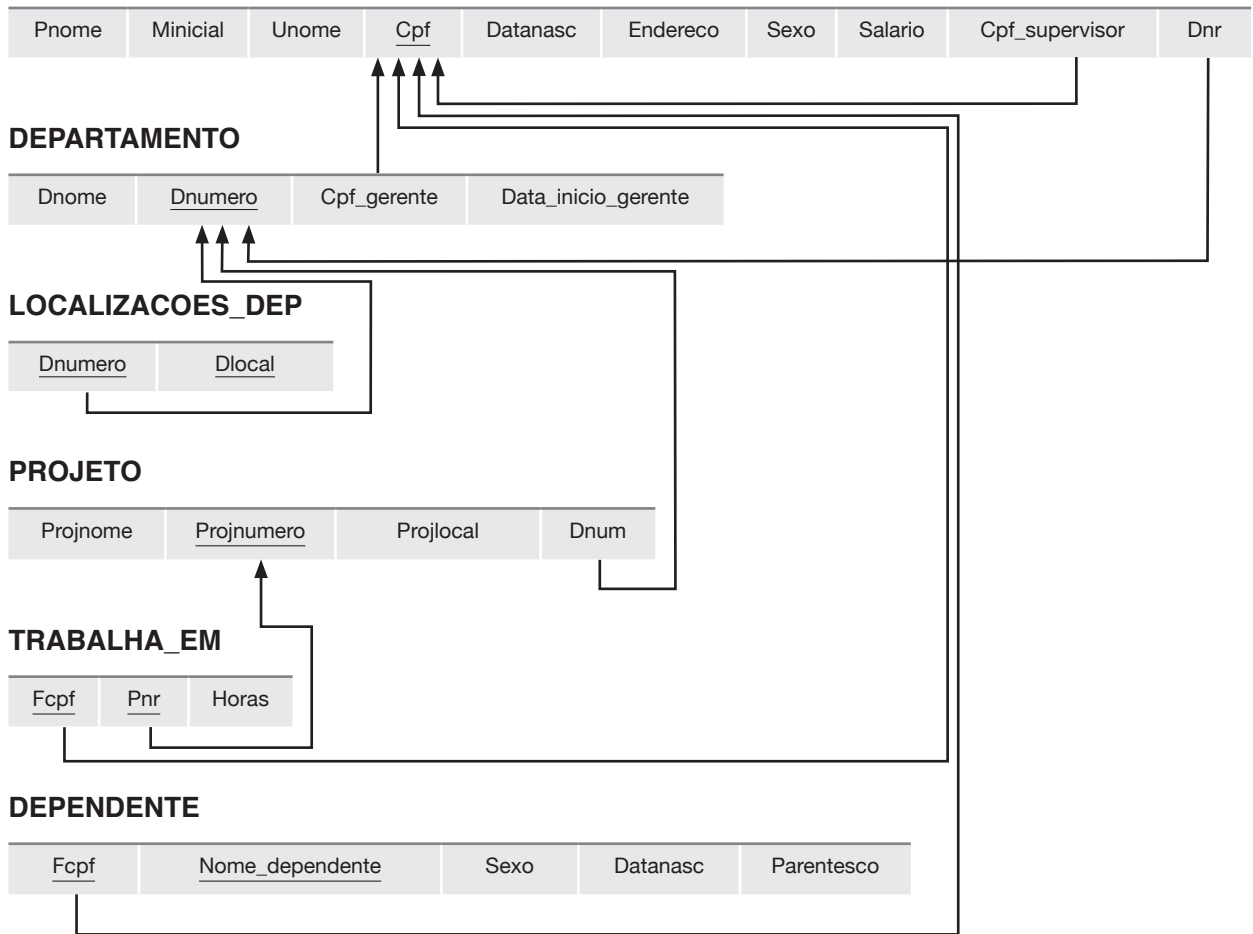


Figura 3.7

Restrições de integridade referencial exibidas no esquema de banco de dados relacional EMPRESA.

gura 3.5 com as restrições de integridade referencial mostradas dessa maneira.

Todas as restrições de integridade deverão ser especificadas no esquema de banco de dados relacional (ou seja, definidas como parte de sua definição) se quisermos impor essas restrições sobre os estados do banco de dados. Logo, a DDL inclui meios para especificar os diversos tipos de restrições de modo que o SGBD possa impô-las automaticamente. A maioria dos SGBDs relacionais admite restrições de chave, integridade de entidade e integridade referencial. Essas restrições são especificadas como uma parte da definição de dados na DDL.

3.2.5 Outros tipos de restrições

As restrições de integridade anteriores estão incluídas na linguagem de definição de dados porque ocorrem na maioria das aplicações de banco de dados. No entanto, elas não incluem uma grande classe de restrições gerais, também chamadas de *restrições*

de integridade semântica, que podem ter de ser especificadas e impostas em um banco de dados relacional. Alguns exemplos dessas restrições são *o salário de um funcionário não deve ser superior ao salário de seu supervisor* e *o número máximo de horas que um funcionário pode trabalhar em todos os projetos por semana é 56*. Essas restrições podem ser especificadas e impostas em programas de aplicação que atualizam o banco de dados, ou usando uma **linguagem de especificação de restrição** de uso geral. Mecanismos chamados **triggers** (gatilhos) e **assertions** (afirmações) podem ser usados. Em SQL, os comandos CREATE ASSERTION e CREATE TRIGGER podem ser usados para essa finalidade (ver Capítulo 5). É mais comum verificar esses tipos de restrições em programas de aplicação do que usar linguagens de especificação de restrição, pois estas às vezes são difíceis e complexas de se usar, conforme discutiremos na Seção 26.1.

Outro tipo de restrição é a de *dependência funcional*, que estabelece um relacionamento funcional

entre dois conjuntos de atributos X e Y . Essa restrição especifica que o valor de X determina um valor exclusivo de Y em todos os estados de uma relação; isso é indicado como uma dependência funcional $X \rightarrow Y$. Usaremos dependências funcionais e outros tipos de dependências nos capítulos 15 e 16 como ferramentas para analisar a qualidade dos projetos relacionais e 'normalizar' relações para melhorar sua qualidade.

Os tipos de restrições que discutimos até aqui podem ser chamados de **restrições de estado**, pois definem as restrições às quais um *estado válido* do banco de dados precisa satisfazer. Outro tipo de restrição, chamadas **restrições de transição**, pode ser definido para lidar com mudanças de estado no banco de dados.¹¹ Um exemplo de uma restrição de transição é: 'o salário de um funcionário só pode aumentar'. Tais restrições costumam ser impostas pelos programas de aplicação ou especificadas usando regras ativas e triggers, conforme discutiremos na Seção 26.1.

3.3 Operações de atualização, transações e tratamento de violações de restrição

As operações do modelo relacional podem ser categorizadas em *recuperações* e *atualizações*. As operações da álgebra relacional, que podem ser usadas para especificar **recuperação**, serão discutidas com detalhes no Capítulo 6. Uma expressão da álgebra relacional forma uma nova relação após a aplicação de uma série de operadores algébricos a um conjunto existente de relações; seu uso principal é consultar um banco de dados a fim de recuperar informações. O usuário formula uma consulta que especifica os dados de interesse, e uma nova relação é formada aplicando operadores relacionais para recuperar esses dados. Esta **relação resultado** torna-se a resposta para a (ou resultado da) consulta do usuário. O Capítulo 6 também introduz a linguagem chamada cálculo relacional, que é usada para definir a nova relação de forma declarativa sem dar uma ordem específica das operações.

Nesta seção, concentramo-nos nas operações de **modificação** ou **atualização** do banco de dados. Existem três operações básicas que podem mudar os estados das relações no banco de dados: Inserir, Excluir e Alterar (ou Modificar). Elas inserem novos dados, excluem dados antigos ou modificam registros de dados existentes. **Insert** é usado para inserir uma ou mais novas tuplas em uma relação, **Delete** é usado para excluir

tuplas, e **Update** (ou **Modify**) é usado para alterar os valores de alguns atributos nas tuplas existentes. Sempre que essas operações são aplicadas, as restrições de integridade especificadas sobre o esquema de banco de dados relacional não devem ser violadas. Nesta seção, discutimos os tipos de restrições que podem ser violadas por cada uma dessas operações e os tipos de ações que podem ser tomados se uma operação causar uma violação. Usamos o banco de dados mostrado na Figura 3.6 para os exemplos e discutimos apenas as restrições de chave, restrições de integridade de entidade e as restrições de integridade referencial, mostradas na Figura 3.7. Para cada tipo de operação, damos alguns exemplos e discutimos as restrições que cada operação pode violar.

3.3.1 A operação Inserir

A operação **Inserir** oferece uma lista de valores de atributo para que uma nova tupla t possa ser inserida em uma relação R . Ela pode violar qualquer um dos quatro tipos de restrições discutidos na seção anterior. As restrições de domínio podem ser violadas se for dado um valor de atributo que não aparece no domínio correspondente ou não é do tipo de dado apropriado. As restrições de chave podem ser violadas se um valor de chave na nova tupla t já existir em outra tupla na relação $r(R)$. A integridade de entidade pode ser violada se qualquer parte da chave primária da nova tupla t for NULL. A integridade referencial pode ser violada se o valor de qualquer chave estrangeira em t se referir a uma tupla que não existe na relação referenciada. Aqui estão alguns exemplos para ilustrar essa discussão.

■ Operação:

Inserir <'Cecilia', 'F', 'Ribeiro', NULL, '05-04-1960', 'Rua Esmeraldas, 35, Bueno Brandão, MG', F, 28.000, NULL, 4> em FUNCIONARIO.

Resultado: Esta inserção viola a restrição de integridade de entidade (NULL para a chave primária Cpf), de modo que é rejeitada.

■ Operação:

Inserir <'Alice', 'J', 'Zelaya', '99988777767', '05-04-1960', 'Rua Souza Lima, 35, Curitiba, PR', F, 28.000, '98765432168', 4> em FUNCIONARIO.

Resultado: Esta inserção viola a restrição de chave porque outra tupla com o mesmo valor de Cpf já existe na relação FUNCIONARIO, e, portanto, é rejeitada.

¹¹ As restrições de estado também podem ser chamadas de **restrições estáticas**, e as restrições de transição também são chamadas de **restrições dinâmicas**.