

Abordagem Relacional

Este capítulo apresenta uma introdução sucinta ao modelo de dados que é usado nos sistemas de gerência de banco de dados do tipo relacional. Não é uma introdução completa à abordagem relacional. É apresentado apenas um conjunto mínimo de conceitos, com o objetivo de permitir que o leitor compreenda o projeto de bancos de dados relacionais, que é discutido nos próximos capítulos. Especificamente, o capítulo detalha como um banco de dados relacional é organizado (que estruturas de dados são usadas, como elas estão relacionadas), mas não discute como um banco de dados relacional pode ser modificado ou acessado, ou seja não apresenta as linguagens de manipulação de dados, como por exemplo, SQL. Para maiores detalhes sobre sistemas de BD relacionais, o leitor deve procurar livros específicos (ver a bibliografia deste capítulo).

Além dos SGBD relacionais, existem outros tipos de sistemas no mercado. Entretanto, hoje, há um claro predomínio dos SGBD relacionais, principalmente fora das plataformas de grande porte. Mesmo nestes ambientes, os SGBD relacionais estão gradativamente substituindo os SGBD de outras abordagens (hierárquica, rede, sistemas proprietários). Além disso, muitos conceitos usados no projeto de BD, como o conceito de normalização, foram criados em combinação com a abordagem relacional. Por esses motivos, vamos considerar unicamente a abordagem relacional neste livro.

4.1 COMPOSIÇÃO DE UM BANCO DE DADOS RELACIONAL

Um banco de dados relacional é composto de *tabelas* ou *relações*. A terminologia *tabela* é mais comum nos produtos comerciais e na prática. Já a terminologia *relação* foi utilizada na literatura original sobre a abordagem relacional (daí a denominação “relacional”) e é mais comum na área acadêmica e nos livros-texto. Neste livro, preferimos adotar a terminologia usada na prática. Entretanto, sempre que apresentarmos um novo conceito, citaremos, entre parênteses, também a terminologia acadêmica.

4.1.1 Tabelas

Uma tabela é um conjunto não ordenado de *linhas* (tuplas, na terminologia acadêmica). Cada linha é composta por uma série de *campos* (valor de atributo, na terminologia acadêmica).

Cada campo é identificado por *nome de campo* (nome de atributo, na terminologia acadêmica). O conjunto de campos das linhas de uma tabela que possuem o mesmo nome formam uma *coluna*.

Emp			
CódigoEmp	Nome	CodigoDepto	CategFuncional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	—

Comparando uma tabela de um banco de dados relacional com um arquivo convencional do sistema de arquivos de um computador, identificam-se as seguintes diferenças:

- ❑ As linhas de uma tabela *não estão ordenadas*. A ordem de recuperação pelo SGBD é arbitrária, a menos que a instrução de consulta tenha especificado explicitamente uma ordenação. Não é possível referenciar linhas de uma tabela por posição. Já em arquivos convencionais, o programador tem controle sobre a ordem de armazenamento e pode referenciar registros por sua posição relativa dentro do arquivo.
- ❑ Os valores de campo de uma tabela são *atômicos* e *mono-valorados*. Em arquivos convencionais, campos podem ser compostos por outros campos (“itens de grupo”) e campos podem ser multi-valorados (“arrays” de Pascal, ou grupos repetidos de COBOL).
- ❑ As linguagens de consulta a bases de dados relacionais permitem o *acesso por quaisquer critérios* envolvendo os campos de uma ou mais linhas. Em arquivos convencionais, para buscar registros com base em valores de seus campos de forma rápida é usualmente necessário que exista algum

tipo de *caminho de acesso*. Um caminho de acesso é uma estrutura auxiliar, como um índice ou uma cadeia de ponteiros, que acelera a recuperação de registros por determinados critérios, evitando a leitura exaustiva de todos registros de um arquivo. Caminhos de acesso também existem em bancos de dados relacionais, mas não são visíveis pelos programadores, isto é, os programadores escrevem consultas à base de dados sem considerar a existência ou não de caminhos de acesso.

4.1.2 Chaves

O conceito básico para estabelecer relações entre linhas de tabelas de um banco de dados relacional é o da *chave*. Em um banco de dados relacional, há ao menos três tipos de chaves a considerar: a chave *primária*, a chave *alternativa*, e a chave *estrangeira*.

4.1.2.1 Chave primária

Uma chave primária é uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela. Na tabela [Emp](#) da Figura 4.1, a chave primária é a coluna [CódigoEmp](#). A Figura 4.2 apresenta um exemplo de uma tabela ([Dependente](#)) que possui uma chave primária *composta* (colunas [CódigoEmp](#) e [NoDepen](#)). Neste caso, apenas um dos valores dos campos que compõem a chave não é suficiente para distinguir uma linha das demais, já que tanto um código de empregado ([CódigoEmp](#)) pode aparecer em diferentes linhas, quanto um número de dependente ([NoDepen](#)) pode aparecer em diferentes linhas. É necessário considerar ambos valores ([CódigoEmp](#) e [NoDepen](#)) para identificar uma linha na tabela, ou seja para identificar um dependente.

Dependente

CódigoEmp	NoDepen	Nome	Tipo	DataNasc
E1	01	João	Filho	12/12/91
E1	02	Maria	Esposa	01/01/50
E2	01	Ana	Esposa	05/11/55
E5	01	Paula	Esposa	04/07/60
E5	02	José	Filho	03/02/85

Pela definição acima, na tabela da Figura 4.2, qualquer combinação de colunas que contenha as colunas **CódigoEmp** e **NoDepen** é uma chave primária. Por isso, nas definições formais de chave primária, exige-se que essa seja *mínima*. Uma chave é mínima quando todas suas colunas forem efetivamente necessárias para garantir o requisito de unicidade de valores da chave. Exemplificando, alguém poderia considerar a combinação de colunas **CódigoEmp**, **NoDepen** e **Tipo** como sendo uma chave primária. Entretanto, se eliminarmos, desta combinação a coluna **Tipo** continuamos frente a uma chave primária. Portanto, a combinação de colunas **CódigoEmp**, **NoDepen** e **Tipo** não obedece o princípio da minimalidade e não deve ser considerada uma chave.

Cabe salientar que, na abordagem relacional, o termo “chave” é empregado com uma conotação diferente daquela usada na área de organização de

arquivos e em alguns sistemas operacionais. Em arquivos convencionais, entende-se por chave qualquer coluna sobre a qual será definido um índice ou algum outro tipo de estrutura de acesso. Na abordagem relacional, ao definir uma chave primária, não está se definindo nenhum caminho de acesso. Está se definindo apenas uma *restrição de integridade*, isto é uma regra que deve ser obedecida em todos estados válidos da BD. No caso da chave primária, a regra é a de unicidade de valores nas colunas que compõem a chave.

4.1.2.2 Chave estrangeira

Uma *chave estrangeira* é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma tabela. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional. No banco de dados da Figura 4.3, a coluna **CodigoDepto** da tabela **Emp** é uma chave estrangeira em relação a chave primária da tabela **Dept**. Isso significa que, na tabela **Emp**, não podem aparecer linhas que contenham um valor do campo **CodigoDepto** que não exista na coluna de mesmo nome da tabela **Emp**. A interpretação desta restrição é que todo empregado deve estar associado a um departamento.

Dept

CodigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

Emp

CodigoEmp	Nome	CodigoDepto	CategFuncional	CIC
E1	Souza	D1	-	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88

A existência de uma chave estrangeira impõe restrições que devem ser garantidas em diversas situações de alteração do banco de dados:

- ☐ *Quando da inclusão de uma linha na tabela que contém a chave estrangeira*
Neste caso, deve ser garantido que o valor da chave estrangeira apareça na coluna da chave primária referenciada. No caso do exemplo da Figura 4.3, isso significa que um novo empregado deve atuar em um departamento já existente no banco de dados.
- ☐ *Quando da alteração do valor da chave estrangeira*
Deve ser garantido que o novo valor de uma chave estrangeira apareça na coluna da chave primária referenciada.
- ☐ *Quando da exclusão de uma linha da tabela que contém a chave primária referenciada pela chave estrangeira*

Deve ser garantido que na coluna chave estrangeira não apareça o valor da chave primária que está sendo excluída. No caso do exemplo da Figura 4.3, isso significa que um departamento não pode ser excluído, caso nele ainda existirem empregados.

A palavra “estrangeira” usada para denominar este tipo de chave pode ser enganosa. Ela pode levar a crer que a chave estrangeira sempre referencia uma chave primária de *outra* tabela. Entretanto, esta restrição não existe. Uma chave primária pode referenciar a chave primária da própria tabela, como mostra a Figura 4.4. Nesta tabela, a coluna **CódigoEmpGerente** é o código de um outro empregado, o gerente do empregado correspondente a linha em questão. Como todo gerente é ele mesmo também um empregado, existe a restrição de que todo valor da coluna **CódigoEmpGerente** deve aparecer na coluna **CódigoEmp**. Assim, a coluna **CódigoEmpGerente** é chave estrangeira em relação a chave primária da própria tabela **Emp**.

Emp

CódigoEmp	Nome	CódigoDepto	CódigoEmpGerente
E5	Souza	D1	—
E3	Santos	D2	E5
E2	Silva	D1	E5
E1	Soares	D1	E1

chave estrangeira:
referencia a chave primária
da própria tabela

4.1.2.3 Chave alternativa

Em alguns casos, mais de uma coluna ou combinações de colunas podem servir para distinguir uma linha das demais. Uma das colunas (ou combinação de colunas) é escolhida como chave primária. As demais colunas ou combinações são denominadas chaves *alternativas*. A Figura 4.5 mostra um exemplo de uma tabela com dados de empregados (**Emp**) na qual tanto a coluna **CódigoEmp** quanto a coluna **CIC** podem ser usadas para distinguir uma linha das demais. Nesta tabela, como a coluna **CódigoEmp** foi escolhida como chave primária, diz-se que a coluna **CIC** é uma chave alternativa.

Emp

CódigoEmp	Nome	CódigoDepto	CategFuncional	CIC
E5	Souza	D1	C5	132.121.331-20
E3	Santos	D2	C5	891.221.111-11
E2	Silva	D1	C2	341.511.773-45
E1	Soares	D1	—	631.692.754-88

Figura 4.5: Chave alternativa (coluna CIC)

Quando, em uma tabela, mais de uma coluna ou combinações de colunas podem servir para distinguir uma linha das demais, surge a questão de que critério deve-se usar para determinar qual das possíveis colunas (ou com-

binação de colunas) será usada como chave primária. No exemplo da Figura 4.5, a questão é que critério foi usado para preferir a coluna **CódigoEmp** como chave primária e considerar a coluna **CIC** como chave alternativa. Porque **CIC** não foi usado como chave primária e **CódigoEmp** como chave alternativa? No fundo, se considerarmos apenas a tabela em que a coluna aparece, não há diferença entre uma coluna ser chave primária ou alternativa. Em ambos casos, apenas está sendo especificada a unicidade de valores de chave. Entretanto, ao considerarmos chaves estrangeiras, a diferenciação entre chave primária e chave alternativa passa a ser relevante. Quando especificamos que uma chave é primária, estamos especificando, além da unicidade de valores, também o fato de esta coluna ser usada nas chaves estrangeiras que referenciam a tabela em questão. Assim, no caso da tabela da Figura 4.5, estamos especificando que tanto os valores de **CódigoEmp** quanto os valores de **CIC** são únicos e adicionalmente que a coluna **CódigoEmp** será usada nas chaves estrangeiras que referenciam a tabela **Emp**.

4.1.3 Domínios e valores vazios

Quando uma tabela do banco de dados é definida, para cada coluna da tabela, deve ser especificado um conjunto de valores (alfanumérico, numérico,...) que os campos da respectiva coluna podem assumir. Este conjunto de valores é chamado de *domínio da coluna* ou *domínio do campo*.

Além disso, deve ser especificado se os campos da coluna podem estar vazios ("null" em inglês) ou não. Estar vazio indica que o campo não recebeu nenhum valor de seu domínio⁶. Na Figura 4.5, o campo **CategFuncional** da linha correspondente ao empregado de código **E1** está vazio. Isso indica que o empregado **E1** não possui categoria funcional ou que esta ainda não foi informada. Na Figura 4.4, aparece outro exemplo de um campo vazio. No caso, o campo **CodigoEmpGerente** vazio indica que o empregado não possui superior hierárquico. As colunas nas quais não são admitidos valores vazios são chamadas de colunas *obrigatórias*. As colunas nas quais podem aparecer campos vazios são chamadas de colunas *opcionais*.

Normalmente, os SGBD relacional exigem que todas colunas que compõem a chave primária sejam obrigatórias. A mesma exigência não é feita para as demais chaves (ver exemplo de chave estrangeira vazia na Figura 4.4).

4.1.4 Restrições de integridade

Um dos objetivos primordiais de um SGBD é a *integridade de dados*. Dizer que os dados de um banco de dados estão íntegros significa dizer que eles refletem corretamente a realidade representada pelo banco de dados e que são consistentes entre si. Para tentar garantir a integridade de um banco de dados os SGBD oferecem o mecanismo de *restrições de integridade*. Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo pró-

prio SGBD. No caso da abordagem relacional, costuma-se classificar as restrições de integridade nas seguintes categorias:

❑ *Integridade de domínio*

Restrições deste tipo especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o *domínio* da coluna). Nos SGBD relacionais comerciais, é possível usar apenas domínios pré-definidos (número inteiro, número real, alfanumérico de tamanho definido, data, ...). O usuário do SGBD não pode definir domínios próprios de sua aplicação (por exemplo, o domínio dos dias da semana ou das unidades da federação).

❑ *Integridade de vazio*

Através deste tipo de restrição de integridade é especificado se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional). Como já foi mencionado, campos que compõem a chave primária sempre devem ser diferentes de vazio.

❑ *Integridade de chave*

Trata-se da restrição que define que os valores da chave primária e alternativa devem ser únicos.

❑ *Integridade referencial*

É a restrição que define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada.

As restrições dos tipos acima especificados devem ser garantidas automaticamente por um SGBD relacional, isto é, não deve ser exigido que o programador escreva procedimentos para garanti-las explicitamente. Há muitas outras restrições de integridade que não se encaixam em nenhuma das categorias acima e que normalmente não são garantidas pelo SGBD. Essas restrições são chamadas de restrições *semânticas*. Alguns exemplos de restrições deste tipo poderiam ser:

- ❑ Um empregado do departamento denominado “Finanças” não pode ter a categoria funcional “Engenheiro”.
- ❑ Um empregado não pode ter um salário maior que seu superior imediato.

4.2 ESPECIFICAÇÃO DE BANCO DE DADOS RELACIONAL

A especificação de um banco de dados relacional (chamada de *esquema* do banco de dados) deve conter no mínimo a definição do seguinte:

- ❑ Tabelas que formam o banco de dados
- ❑ Colunas que as tabelas possuem
- ❑ Restrições de integridade

Na prática, na definição de esquemas relacionais são usadas diversas notações, que variam de um SGBD para o outro. Nesta seção, vamos apresentar apenas uma notação resumida para modelos lógicos relacionais. Essa notação é incompleta mas compacta, que é útil para exemplos como os mostrados no livro, bem como para discussões sobre a estrutura geral do banco de dados, quando não se deseja entrar no maior nível de detalhamento.

A Figura 4.6 apresenta o esquema correspondente às tabelas da Figura 4.3 usando a notação resumida.

Emp (CodigoEmp, Nome, CodigoDepto, CategFuncional, CIC)
CodigoDept referencia Dept
Dept (CodigoDepto, Nome)

Figura 4.6: Esquema do banco de dados da Figura 4.3

Nesta notação, são listadas as tabelas e, para cada tabela, enumerados, entre parênteses, os nomes das colunas que compõem a tabela. As colunas que compõem a chave primária aparecem sublinhadas. Após a definição da tabela aparecem as definições das chaves estrangeiras que aparecem na tabela na forma:

<nome de coluna ch. estrangeira> referencia <nome de tabela>

quando tratar-se de uma chave estrangeira composta de uma única coluna, ou na forma:

(<nome de coluna>1,<nome de coluna>2,...) referencia <nome de tabela>

quando tratar-se de uma chave estrangeira composta por múltiplas colunas.