

Banco de Dados I

04 - O Modelo Relacional

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

- Atualmente, o modelo de banco de dados relacional é sem sombra de dúvidas o modelo de dados dominante na maioria dos SGBD;
- Um banco de dados relacional é composto de **tabelas** (também chamadas de **relações**)¹;

¹O termo tabela é mais comum nos produtos comerciais e na prática. Já o termo relação foi utilizado na literatura original sobre a abordagem relacional (daí a denominação *relacional*) e é mais comum na área acadêmica e nos livros-texto

Tabelas

- Uma tabela é um conjunto não ordenado de **linhas** (ou **tuplas**);
- Cada linha é composta por uma série de **campos** (ou **valores de atributo**);
- Cada campo é identificado por nome de campo (ou nome de atributo).
O conjunto de campos das linhas de uma tabela que possuem o mesmo nome formam uma **coluna**;

Exemplo: Tabela *Funcionário*

ID	Nome	ID-Depto	ID-Categoria
5	Sousa	D1	C5
3	Santos	D2	C5
2	Silva	D1	C2
1	Soares	D1	-

- 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:
 - Chaves primárias;
 - Chaves alternativas;
 - Chaves estrangeiras.

Chave Primária

- Uma chave primária é uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela;
- Qual seria chave primária da tabela *Funcionário*?

<u>ID</u>	Nome	ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

- E para a tabela *Dependente* a seguir?

ID-Func	Num-Dep	Nome	Tipo	Nascimento
1	1	João	Filho	12/12/2001
1	2	Maria	Esposa	05/12/1980
2	1	Ana	Esposa	06/08/1985
5	1	Paula	Esposa	12/11/1988
5	2	José	Filho	03/04/2010

Chave Primária

- Uma chave primária é uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela;
- Qual seria chave primária da tabela *Funcionário*?

<u>ID</u>	Nome	ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

- E para a tabela *Dependente* a seguir?

ID-Func	Num-Dep	Nome	Tipo	Nascimento
1	1	João	Filho	12/12/2001
1	2	Maria	Esposa	05/12/1980
2	1	Ana	Esposa	06/08/1985
5	1	Paula	Esposa	12/11/1988
5	2	José	Filho	03/04/2010

Chaves Primárias Compostas

<u>ID-Func</u>	<u>Num-Dep</u>	Nome	Tipo	Nascimento
<u>1</u>	<u>1</u>	João	Filho	12/12/2001
<u>1</u>	<u>2</u>	Maria	Esposa	05/12/1980
<u>2</u>	<u>1</u>	Ana	Esposa	06/08/1985
<u>5</u>	<u>1</u>	Paula	Esposa	12/11/1988
<u>5</u>	<u>2</u>	José	Filho	03/04/2010

- A tabela *Dependente* que possui uma chave primária composta (colunas *ID-Func* e *Num-Dep*);
- 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 quanto o número do dependente podem aparecer em diferentes linhas;
- É necessário considerar ambos valores para identificar uma linha na tabela, ou seja para identificar um dependente.

Chaves Estrangeiras

- 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 a seguir, a coluna *ID-Depto* da tabela *Funcionário* (marcada com *) é uma chave estrangeira em relação a chave primária da tabela *Departamento*. Isso significa que, na tabela *Funcionário*, não podem aparecer linhas com um valor em *ID-Depto* que não exista na coluna de mesmo nome da tabela *Departamento*.

Departamento

<u>ID-Depto</u>	Nome-Depto
<u>D1</u>	Compras
<u>D2</u>	Engenharia
<u>D3</u>	Vendas

Funcionário

<u>ID</u>	Nome	*ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

Chaves Estrangeiras

- 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 a seguir, a coluna *ID-Depto* da tabela *Funcionário* (marcada com *) é uma chave estrangeira em relação a chave primária da tabela *Departamento*. Isso significa que, na tabela *Funcionário*, não podem aparecer linhas com um valor em *ID-Depto* que não exista na coluna de mesmo nome da tabela *Departamento*.

Departamento

<u>ID-Depto</u>	Nome-Depto
<u>D1</u>	Compras
<u>D2</u>	Engenharia
<u>D3</u>	Vendas

Funcionário

<u>ID</u>	Nome	*ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

Restrições Impostas pelas Chaves Estrangeiras

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

- **Quando uma linha é incluída ou alterada 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. Em nosso exemplo, isso significa que um novo funcionário deve atuar em um departamento já existente no banco de dados.

- **Quando uma linha é excluída na tabela que contém a chave primária referenciada pela chave estrangeira.²**

Deve ser garantido que na coluna chave estrangeira não apareça o valor da chave primária que está sendo excluída. Em nosso exemplo, isso significa que um departamento não pode ser excluído, caso nele ainda existirem empregados.

²Posteriormente, veremos que esta restrição pode ser contornada

Chaves Estrangeiras em uma Única Tabela

- A palavra *chave estrangeira* pode ser enganosa, levando a acreditar que a chave estrangeira sempre referencia uma chave primária de outra tabela;
- Entretanto, esta restrição não existe. Uma chave estrangeira pode referenciar a chave primária da própria tabela.

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria	*ID-Supervisor
<u>5</u>	Sousa	D1	C5	-
<u>3</u>	Santos	D2	C5	5
<u>2</u>	Silva	D1	C2	5
<u>1</u>	Soares	D1	-	2

Neste exemplo, a coluna *ID-Supervisor* é o código do funcionário supervisor.

Chaves Alternativas

- 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.

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria	<u>CPF</u>
<u>5</u>	Sousa	D1	C5	<u>120.681.015-18</u>
<u>3</u>	Santos	D2	C5	<u>657.761.408-45</u>
<u>2</u>	Silva	D1	C2	<u>181.689.155-05</u>
<u>1</u>	Soares	D1	-	<u>806.321.034-10</u>

Neste exemplo, a coluna *CPF* é uma chave alternativa porque podemos distinguir um funcionário dos demais através de seu CPF.

Chaves Primárias x Chaves Alternativas

- Há duas diferenças básicas entre chaves primárias e alternativas:
 - *Chaves primárias* podem ser usadas para a criação de chaves estrangeiras;
 - *Chaves alternativas* permitem valores nulos³.

Qual será a chave primária?

- 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 combinação de colunas) será usada como chave primária;
- Na tabela *Funcionário*, que critério foi usado para preferir a coluna *ID* como chave primária e considerar a coluna *CPF* como chave alternativa?
- Poderia ser qualquer uma, mas o campo *ID* é mais curto. Qual a vantagem disto?

³Mais adiante veremos o que são estes valores

Chaves Primárias x Chaves Alternativas

- Há duas diferenças básicas entre chaves primárias e alternativas:
 - *Chaves primárias* podem ser usadas para a criação de chaves estrangeiras;
 - *Chaves alternativas* permitem valores nulos³.

Qual será a chave primária?

- 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 combinação de colunas) será usada como chave primária;
- Na tabela *Funcionário*, que critério foi usado para preferir a coluna *ID* como chave primária e considerar a coluna *CPF* como chave alternativa?
- Poderia ser qualquer uma, mas o campo *ID* é mais curto. Qual a vantagem disto?

³Mais adiante veremos o que são estes valores

Chaves Primárias x Chaves Alternativas

- Há duas diferenças básicas entre chaves primárias e alternativas:
 - *Chaves primárias* podem ser usadas para a criação de chaves estrangeiras;
 - *Chaves alternativas* permitem valores nulos³.

Qual será a chave primária?

- 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 combinação de colunas) será usada como chave primária;
- Na tabela *Funcionário*, que critério foi usado para preferir a coluna *ID* como chave primária e considerar a coluna *CPF* como chave alternativa?
- Poderia ser qualquer uma, mas o campo *ID* é mais curto. Qual a vantagem disto?

³Mais adiante veremos o que são estes valores

Domínios

- Quando uma tabela do banco de dados é definida, para cada coluna da tabela, deve ser especificado um conjunto de valores que os campos da respectiva coluna podem assumir;
- Este conjunto de valores é chamado de domínio da coluna ou domínio do campo (ou ainda tipo). Além disso, deve ser especificado se os campos da coluna podem estar vazios (ser nulos ou *null*). Estar vazio indica que o campo não recebeu nenhum valor de seu domínio. Exemplo:

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

O campo *ID-Categoria* do funcionário 1 apresenta valor nulo.

Valores Obrigatórios

- 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 relacionais exigem que todas colunas que compõem a chave primária sejam obrigatórias.

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;
- Os SGBD oferecem mecanismos de *restrições de integridade* para que um banco de dados fique íntegro;
- Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo próprio SGBD. No caso da abordagem relacional, costuma-se classificar as restrições de integridade nas categorias:
 - Integridade de domínio;
 - Integridade de vazio;
 - Integridade de chave;
 - Integridade referencial.

Categorias de Restrições de Integridade

Integridade de domínio especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna);

Integridade de vazio especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional);

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.

Categorias de Restrições de Integridade

Integridade de domínio especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna);

Integridade de vazio especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional);

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.

Categorias de Restrições de Integridade

- Integridade de domínio** especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna);
- Integridade de vazio** especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional);
- 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.

Categorias de Restrições de Integridade

- Integridade de domínio** especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna);
- Integridade de vazio** especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional);
- 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.

Restrições Semânticas

- As categorias de restrições mencionadas são garantidas automaticamente pelo SGBD relacional, isto é, não precisam ser tratadas via programação;
- Há muitas outras restrições de integridade que não se encaixam em nenhuma das categorias mencionadas e que normalmente não são garantidas pelo SGBD. Essas restrições são chamadas de restrições *semânticas*;
- Exemplos de restrições semânticas:
 - 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.

Modelagem Lógica

- Na modelagem conceitual vimos como representar um banco de dados através de DER, sem nos preocuparmos com o modelo relacional ou com SGBD específicos;
- Na modelagem lógica vamos especificar bancos de dados considerando o modelo relacional, com isto os modelos de dados (também chamados de esquemas lógicos) criados estarão mais próximos do esquema físico dos SGBD relacionais;
- A especificação de um banco de dados relacional deve conter:
 - Tabelas que formam o banco de dados;
 - Colunas que as tabelas possuem;
 - Restrições de integridade.
- Esquemas lógicos podem ser representados textualmente e graficamente através de diagramas;

Representação Textual

- O esquema lógico pode ser escrito diretamente com a linguagem *Structured Query Language (SQL)*, porém vamos utilizar uma notação mais simplificada para facilitar o entendimento e, posteriormente, converter para SQL. Exemplo:

Funcionário

ID	Nome	ID-Depto	ID-Categoria
5	Sousa	D1	C5
3	Santos	D2	C5
2	Silva	D1	C2
1	Soares	D1	-

```
funcionario(id, nome, id_depto, id_categoria);
```

- Os nomes no esquema lógico devem ser parecidos com nomes de variáveis, ou seja, não devem conter caracteres especiais.

- No esquema lógico devemos considerar domínios de campos que serão utilizados em SGBD;
- Existem diversos domínios padrões da SQL e cada SGBD pode possuir domínio específicos, porém inicialmente vamos considerar os seguintes:
 - CHAR** Valores alfanuméricos com tamanho fixo;
 - DATE** Datas;
 - DATETIME** Data e hora;
 - FLOAT** Valores numéricos fracionários;
 - INTENGER** Valores numéricos inteiros (pode ser abreviado para **INT**);
 - TIME** Hora;
 - VARCHAR** Valores alfanuméricos com tamanho variável;

- Para a tabela funcionário podemos considerar os seguintes domínios:

```
funcionario(id INT, nome VARCHAR(50), id_depto CHAR(4),  
            id_categoria CHAR(4));
```

- Para os tipos CHAR e VARCHAR é preciso informar o número máximo de caracteres a ser armazenado;
- Se os campos ID, ID-Depto e ID-Categoria fossem numéricos poderíamos utilizar o tipo INT.

Integridade de Vazio

- A integridade de vazio pode ser feita com o sinal de + antes do campo, neste caso os campos precedidos com este sinal não podem aceitar valores nulos. Exemplo:

Funcionário

ID	Nome	ID-Depto	ID-Categoria
5	Sousa	D1	C5
3	Santos	D2	C5
2	Silva	D1	C2
1	Soares	D1	-

```
funcionario(+id INT, +nome VARCHAR(50),  
            +id_depto CHAR(4), id_categoria CHAR(4));
```

- Estamos considerando que os campos ID, Nome e ID-Depto são obrigatórios.

Chaves Primárias

- A representação da chave primária é feita mantendo os campos que a compõem sublinhados;
- Exemplo:

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

```
funcionario(+id INT, +nome VARCHAR(50),  
            +id_depto CHAR(4), id_categoria CHAR(4));
```

Chaves Alternativas

- As chaves alternativas permanecem com o sublinhado duplo;
- Exemplo:

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria	<u>CPF</u>
<u>5</u>	Sousa	D1	C5	<u>120.681.015-18</u>
<u>3</u>	Santos	D2	C5	<u>657.761.408-45</u>
<u>2</u>	Silva	D1	C2	<u>181.689.155-05</u>
<u>1</u>	Soares	D1	-	<u>806.321.034-10</u>

```
funcionario(+id INT, +nome VARCHAR(50),  
           +id_depto CHAR(4), id_categoria CHAR(4));  
           cpf CHAR(11))
```

Chaves Estrangeiras

- As colunas que compõem a chave estrangeira devem ser marcadas com * e após a definição da tabela especificadas. Exemplo:

<u>ID-Depto</u>	Nome-Depto
<u>D1</u>	Compras
<u>D2</u>	Engenharia
<u>D3</u>	Vendas

```
departamento( +id_depto CHAR(4),  
               +nome_depto VARCHAR(30))
```

<u>ID</u>	Nome	*ID-Depto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

```
funcionario(+id INT, +nome VARCHAR(50),  
            *+id_depto CHAR(4), id_categoria CHAR(4));  
funcionario.id_depto: departamento.id_depto
```

- As representações gráficas de esquema lógico são muito utilizadas em ferramentas de modelagem de dados;
- Cada ferramenta utiliza uma notação própria, contudo as diversas notações são bastante similares;
- Vamos utilizar a notação da ferramenta *SQL Power Architect Community Edition* disponível em <http://code.google.com/p/power-architect>.

Exemplo

Funcionário

<u>ID</u>	Nome	ID-Depto	ID-Categoria	<u>CPF</u>
<u>5</u>	Sousa	D1	C5	<u>120.681.015-18</u>
<u>3</u>	Santos	D2	C5	<u>657.761.408-45</u>
<u>2</u>	Silva	D1	C2	<u>181.689.155-05</u>
<u>1</u>	Soares	D1	-	<u>806.321.034-10</u>

funcionario

id: INTEGER [PK]

nome: VARCHAR(50)

id_depto: CHAR(4)

id_categoria: CHAR(4)

cpf: CHAR(11) [AK]

```
funcionario(+id INT,  
            +nome VARCHAR(50),  
            +id_depto CHAR(4),  
            id_categoria CHAR(4));  
            cpf CHAR(11)  
)
```

funcionario

id: INTEGER [PK]

nome: VARCHAR(50)

id_depto: CHAR(4)

id_categoria: CHAR(4)

cpf: CHAR(11) [AK]

```
funcionario(+id INT,  
            +nome VARCHAR(50),  
            +id_depto CHAR(4),  
            id_categoria CHAR(4));  
cpf CHAR(11)  
)
```

- A notação da ferramenta SQL Power Architect não exibe a integridade de vazio, mas é possível defini-la na ferramenta.

Chaves Estrangeiras

<u>ID-Deppto</u>	Nome-Deppto
<u>D1</u>	Compras
<u>D2</u>	Engenharia
<u>D3</u>	Vendas

<u>ID</u>	Nome	*ID-Deppto	ID-Categoria
<u>5</u>	Sousa	D1	C5
<u>3</u>	Santos	D2	C5
<u>2</u>	Silva	D1	C2
<u>1</u>	Soares	D1	-

funcionario

id: INTEGER [PK]
nome: VARCHAR(50)
id_categoria: CHAR(4)
cpf: CHAR(11) [AK]
id_depto: CHAR(4) [FK]

departamento

id_depto: CHAR(4) [PK]
nome_depto: VARCHAR(30)



Referências



(2010).

Sql power architect.



Elmasri, R. and Navathe, S. B. (2011).

Sistemas de banco de dados.

Pearson Addison Wesley, São Paulo, 6 edition.



Heuser, C. A. (2009).

Projeto de banco de dados.

Bookman, Porto Alegre, 6 edition.



Ramakrishnan, R. and Gehrke, J. (2008).

Sistemas de gerenciamento de banco de dados.

McGrawHill, São Paulo, 3 edition.