

**Instituto Federal Minas Gerais - Campus Bambuí**  
**Departamento de Engenharia e Computação - DEC**  
**Curso de Engenharia de Computação - ENGCAMP**

# **Banco de Dados I**

## **04 - O Modelo Relacional**

Marcos Roberto Ribeiro

2024



- Atualmente, o modelo de banco de dados relacional é o modelo de dados dominante na maioria dos SGBD
- Um banco de dados relacional é composto de **tabelas** (também chamadas de **relações**)
- O termo tabela é mais comum nos produtos comerciais
- Já o termo relação é mais comum na área acadêmica, sua origem está na abordagem relacional

# 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	IDDepto	IDCategoria
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 três tipos de chaves:
  - 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
- Quais as chaves primárias das tabelas a seguir?

Funcionário

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

Dependente

IDFunc	NumDep	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
- Quais as chaves primárias das tabelas a seguir?

Funcionário

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

Dependente

IDFunc	NumDep	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>IDFunc</u>	<u>NumDep</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* possui uma chave primária composta (colunas *IDFunc* e *NumDep*)
- Nesse 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 *IDDepto* 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 *IDDepto* que não exista na coluna de mesmo nome da tabela *Departamento*

Exemplo: Departamento

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

Exemplo: Funcionário

<u>ID</u>	Nome	*IDDepto	IDCategoria
<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 *IDDepto* 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 *IDDepto* que não exista na coluna de mesmo nome da tabela *Departamento*

Exemplo: Departamento

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

Exemplo: Funcionário

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

# 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.**
    - Nesse 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.**
    - Deve ser garantido que na coluna da 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 existam empregados

# Chaves estrangeiras em uma única tabela

- A palavra *chave estrangeira* pode ser enganosa, levando a acreditar que a referencia sempre seja uma chave primária de outra tabela
- Entretanto, uma chave estrangeira pode referenciar a chave primária da própria tabela

## Exemplo: Funcionário

<u>ID</u>	Nome	IDDepto	IDCategoria	*IDSupervisor
<u>5</u>	Sousa	D1	C5	-
<u>3</u>	Santos	D2	C5	<b>5</b>
<u>2</u>	Silva	D1	C2	<b>5</b>
<u>1</u>	Soares	D1	-	<b>2</b>

Nesse exemplo, a coluna *IDSupervisor* 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

## Exemplo: Funcionário

<u>ID</u>	Nome	IDDepto	IDCategoria	<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>

Nesse 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 nulos<sup>1</sup>

## 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 qual critério deve ser usado para escolher a 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?

<sup>1</sup>Os valores nulos indicam que o campo não foi preenchido com nenhum valor

# 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 nulos<sup>1</sup>

## 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 qual critério deve ser usado para escolher a 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?

<sup>1</sup>Os valores nulos indicam que o campo não foi preenchido com nenhum valor

# 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 nulos<sup>1</sup>

## 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 qual critério deve ser usado para escolher a 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?

<sup>1</sup>Os valores nulos indicam que o campo não foi preenchido com nenhum valor

# 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 podem assumir
- Esse 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

## Funcionário

<u>ID</u>	Nome	IDDepto	IDCategoria
<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 *IDCategoria* do funcionário 1 apresenta valor nulo





- 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*
- Todas as colunas que compõem a chave primária devem ser 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
- Existem mecanismos de *restrições de integridade* para que um banco de dados permaneça íntegro
- Uma restrição de integridade é uma regra de consistência de dados que deve ser garantida
- No caso da abordagem relacional, as restrições de integridade são classificadas como:
  - Integridade de domínio
  - Integridade de vazio
  - Integridade de chave
  - Integridade referencial

# Categorias de restrições de integridade

---



**Integridade de domínio:** especifica que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna)

**Integridade de vazio:** define se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional)

**Integridade de chave:** especifica que os valores da chave primária e alternativa devem ser únicos

**Integridade referencial:** 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:** especifica que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna)

**Integridade de vazio:** define se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional)

**Integridade de chave:** especifica que os valores da chave primária e alternativa devem ser únicos

**Integridade referencial:** 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:** especifica que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna)

**Integridade de vazio:** define se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional)

**Integridade de chave:** especifica que os valores da chave primária e alternativa devem ser únicos

**Integridade referencial:** 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:** especifica que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (o domínio da coluna)

**Integridade de vazio:** define se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional)

**Integridade de chave:** especifica que os valores da chave primária e alternativa devem ser únicos

**Integridade referencial:** define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada

- As categorias de restrições mencionadas são garantidas automaticamente pelo sistema de banco de dados, 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 sistema de banco de dados
- 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



- Na modelagem conceitual, vimos como representar um banco de dados através de DER, sem nos preocuparmos com o modelo relacional
- 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 sistemas de banco de dados 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	IDDepto	IDCategoria
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 sistemas de bancos de dados
- Existem diversos domínios padrões da SQL e cada sistema pode possuir domínio específicos.
- Porém, inicialmente, vamos considerar os seguintes:

CHAR: Valores alfanuméricos com tamanho fixo

VARCHAR: Valores alfanuméricos com tamanho variável

INTEGER: Valores numéricos inteiros (pode ser abreviado para **INT**)

FLOAT: Valores numéricos fracionários

DATE: Datas

TIME: Hora

DATETIME: Data e hora



- 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 *IDDepto* e *IDCategoria* fossem numéricos, poderíamos utilizar o tipo INT

# Integridade de vazio

- A integridade de vazio pode ser especificada com o sinal de + antes do campo
- Nesse caso, os campos precedidos com esse sinal não podem aceitar valores nulos

## Exemplo: Funcionário

ID	Nome	IDDepto	IDCategoria
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 IDDepto são obrigatórios

# Chaves primárias

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

## Exemplo: Funcionário

<u>ID</u>	Nome	IDDepto	IDCategoria
<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	IDDepto	IDCategoria	<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 \*
- Além disso, é necessário especificar a tabela de origem

## Exemplo: Departamento e funcionário

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

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

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

```
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, mas, em geral, todas as notações têm uma certa semelhança
- Duas ferramentas interessantes são:
  - MySQL Workbench:** exclusivo para o SGBD MySQL com suporte a múltiplos diagramas (<https://www.mysql.com/products/workbench/>)
  - SQL Power Architect:** suporta múltiplos SGBDs, mas os projetos devem ter um único diagrama (<http://www.bestofbi.com/page/architect>)



# Exemplo

## Representação Textual

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

## SQL Power Architect

funcionario

id: INTEGER NOT NULL [ PK ]

nome: VARCHAR(50) NOT NULL

id\_depto: CHAR(4) NOT NULL

id\_categoria: CHAR(4)

cpf: CHAR(11) [ AK ]

## MySQL Workbench



funcionario



id INT



nome VARCHAR(50)



id\_depto CHAR(4)

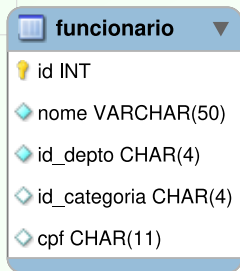


id\_categoria CHAR(4)



cpf CHAR(11)

# Restrição de unicidade no MySQL Workbench



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

- A ferramenta MySQL Workbench não exibe a restrição de unicidade, mas é possível defini-la

# Chaves estrangeiras

## SQL Power Architect

funcionario

id: INTEGER NOT NULL [ PK ]

nome: VARCHAR(50) NOT NULL

id\_categoria: CHAR(4)

cpf: CHAR(11) [ AK ]

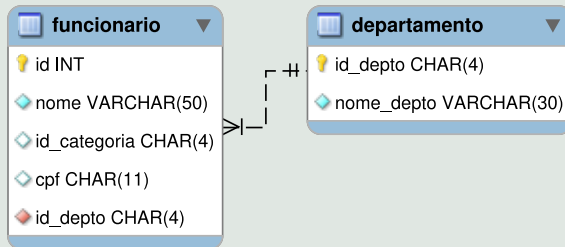
id\_depto: CHAR(4) NOT NULL [ FK ]

departamento

id\_depto: CHAR(4) NOT NULL [ PK ]

nome\_depto: VARCHAR(30) NOT NULL

## MySQL Workbench





ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

HEUSER, C. A. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados**. 3. ed. São Paulo: McGrawHill, 2008.