



*BANCO DE DADOS*

**ESTRUTURA DE UM  
SGBD RELACIONAL**

**Prof. Fabiano Papaiz  
IFRN**

## ESTRUTURA DE UM SGBD RELACIONAL

- Após modelarmos conceitualmente o nosso banco de dados, o próximo passo será construir o *Modelo Lógico* desse banco
- Relembrando:
- O *Modelo Lógico* é uma descrição do banco de dados de forma **dependente da implementação em um SGBD**
- O modelo lógico registra como os dados serão armazenados no SGBD, com sua organização em *tabelas, colunas (ou campos), relacionamentos* etc.

# ESTRUTURA DE UM SGBD RELACIONAL

- Mas primeiramente devemos entender como um **SGBD Relacional** é estruturado
- Estudaremos somente os **SGBDs Relacionais**, que são os mais comumente utilizados para o desenvolvimento da maioria dos sistemas
- Existem outros tipos de SGBD, como por exemplo:
  - **Orientado a Objetos**
  - **NoSQL** (*Not only SQL*) que são orientados a:
    - *Chave-Valor*
    - *Colunas*
    - *Documentos*
    - *Grafos etc*

# ESTRUTURA DE UM SGBD RELACIONAL

- Um *SGBD Relacional*, ou *Banco de Dados Relacional*, é composto por *tabelas* (ou *relações*)
- Uma *tabela* é um conjunto não ordenado de *registros* (ou *linhas* ou *tuplas*) e identificada por um *nome de tabela*
- Cada *registro* é composto por uma série de *campos* (ou *colunas*) e cada *campo* é identificado por um *nome de campo*

○ *Tabela*

○ *Campos (Colunas)*

○ *Registros (Linhas)*

The diagram shows a table titled "Produto" with three columns: "codigo", "descricao", and "valor\_venda". The first two rows of data are highlighted. Arrows point from the text labels on the left to the corresponding parts of the table: "Tabela" points to the table title, "Campos (Colunas)" points to the header row, and "Registros (Linhas)" points to the two data rows.

Produto		
codigo	descricao	valor_venda
1	Computador	2500
2	Impressora	350

# ESTRUTURA DE UM SGBD RELACIONAL

- O conceito básico para estabelecer relacionamentos entre registros de tabelas (mesma tabela ou tabelas diferentes) é o da **chave** (do inglês *key*)
- Os 3 tipos mais comuns de chaves são:
  - **Chave Primária** (*Primary Key*)
  - **Chave Estrangeira** (*Foreign Key*)
  - **Chave Alternativa** (*Alternate Key*)

# ESTRUTURA DE UM SGBD RELACIONAL

**Chave Primária** (*Primary Key*)

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- É um campo ou combinação mínima de campos cujos valores **distinguem um registro dos demais** registros dentro da mesma tabela

<b>Produto</b>		
<b>codigo</b>	<b>descricao</b>	<b>valor_venda</b>
1	Computador	2500
2	Impressora	350

- Neste exemplo, a **chave primária** da tabela **Produto** é o campo **codigo**, pois os valores deste campo **identificam unicamente** cada registro dessa tabela.  
**\*\*Não irão existir 2 produtos com o mesmo código.**

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Qual deve ser a chave primária para a tabela de dependentes dos funcionários exibida abaixo?

<b>Dependente</b>			
<b>cod_funcionario</b>	<b>cod_dependente</b>	<b>nome</b>	<b>tipo</b>
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho



# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Qual deve ser a chave primária para a tabela de dependentes dos funcionários exibida abaixo?

Dependente			
cod_funcionario	cod_dependente	nome	tipo
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho

- Inicialmente já podemos descartar os campos **nome** e **tipo**, pois poderão haver valores repetidos nestas colunas (como *José* e *filho*)

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Qual deve ser a chave primária para a tabela de dependentes dos funcionários exibida abaixo?

Dependente			
cod_funcionario	cod_dependente	nome	tipo
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho

- Também não podemos usar apenas o campo **cod\_funcionario**, pois existem valores repetidos (101 e 103) para linhas diferentes, pois um mesmo funcionário pode ter mais de 1 dependente

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Qual deve ser a chave primária para a tabela de dependentes dos funcionários exibida abaixo?

<b>Dependente</b>			
<b>cod_funcionario</b>	<b>cod_dependente</b>	<b>nome</b>	<b>tipo</b>
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho

- Para o campo **cod\_dependente** também podem existir valores repetidos (1 e 2) para linhas diferentes, pois para cada funcionário este código inicia do valor 1

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Qual deve ser a chave primária para a tabela de dependentes dos funcionários exibida abaixo?

Dependente			
cod_funcionario	cod_dependente	nome	tipo
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho

- **Solução:** usar os campos **cod\_funcionario** e **cod\_dependente** de forma combinada para definir a **chave primária**. Assim, podemos garantir que não irão ocorrer valores repetidos para este conjunto de campos

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Primária** (*Primary Key*)
- Através da chave primária, definimos uma *restrição de integridade de unicidade* para a tabela e, com isso, o SGBD irá nos garantir que não ocorram valores repetidos para os campos que compõem a chave

Produto		
codigo	descricao	valor_venda
1	Computador	2500
2	Impressora	350

Dependente			
cod_funcionario	cod_dependente	nome	tipo
101	1	João	filho
101	2	Maria	esposa
102	1	José	esposo
103	1	Ana	esposa
103	2	José	filho

# ESTRUTURA DE UM SGBD RELACIONAL

**Chave Estrangeira** (*Foreign Key*)

# ESTRUTURA DE UM SGBD RELACIONAL

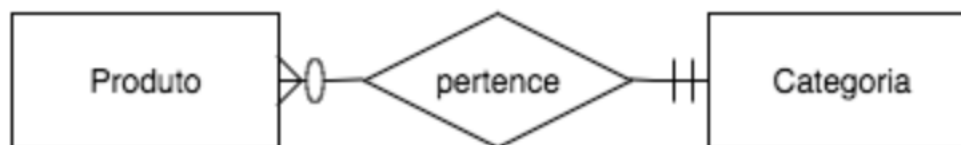
- **Chave Estrangeira** (*Foreign Key*)
- É o mecanismo pelo qual o SGBD Relacional implementa os relacionamentos entre os registros das tabelas, podendo ser entre tabelas diferentes ou dentro de uma mesma tabela
- O campo ou conjunto de campos que compõem a **chave estrangeira devem necessariamente ser chave primária em outra tabela ou dentro da própria tabela**
  - A **chave estrangeira** deve fazer referência a uma **chave primária**

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Estrangeira** (*Foreign Key*)
- Exemplo de chave estrangeira entre **tabelas diferentes**

Produto				Categoria	
<u>codigo</u>	descricao	valor_venda	cod_categoria	<u>codigo</u>	descricao
100	Computador	2500	1	1	Informática
200	Impressora	350	1	2	Eletrônicos
300	Celular	350	2		

Diagram illustrating a foreign key relationship between the **Produto** and **Categoria** tables. The **Produto** table has a foreign key **cod\_categoria** that references the primary key **codigo** in the **Categoria** table. Red arrows indicate the mapping: **Produto** (100, 200) maps to **Categoria** (1), and **Produto** (300) maps to **Categoria** (2). The **Categoria** table is labeled as **chave primária** and the **Produto** table is labeled as **chave estrangeira**.





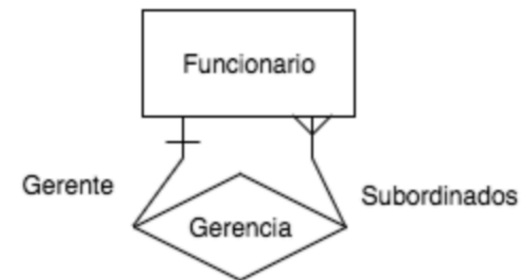
# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Estrangeira** (*Foreign Key*)
- Exemplo de chave estrangeira em uma **mesma tabela**, também conhecido como **auto-relacionamento**

<b>Funcionario</b>		
<u>codigo</u>	nome	cod_func_gerente
1	José	
2	Maria	1
3	Ana	1
4	João	
5	Paula	4
6	Antônio	4

chave primária

chave estrangeira



# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Estrangeira** (*Foreign Key*)
- Impõe as seguintes *Restrições de Integridade* dos dados:
  1. ***Na inclusão de um registro que contém uma chave estrangeira***
    - O valor inserido na chave estrangeira deverá existir em algum registro na tabela da chave primária referenciada
  2. ***Na alteração do valor de uma chave estrangeira***
    - O novo valor da chave estrangeira deverá existir em algum registro na tabela da chave primária referenciada
  3. ***Na exclusão de um registro que contém a chave primária referenciada por uma chave estrangeira***
    - Um registro não poderá ser excluído se a sua chave primária estiver sendo utilizada por uma chave estrangeira

# ESTRUTURA DE UM SGBD RELACIONAL

**Chave Alternativa** (*Alternate Key*)

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Alternativa** (*Alternate Key*)
- Em alguns casos, mais de um campo (ou campos) pode servir para distinguir um registro dos demais existentes em uma tabela
- Assim, teremos mais de uma opção para escolha da chave primária de uma tabela

<b>Aluno</b>		
<b>matricula</b>	<b>nome</b>	<b>cpf</b>
2017001	José	111.222.333-44
2017002	Maria	777.888.999-00
2017003	João	555.666.777-88

**Qual campo devemos utilizar como chave primária?  
Qualquer um deles irá servir!**

# ESTRUTURA DE UM SGBD RELACIONAL

- **Chave Alternativa** (*Alternate Key*)
- Dentre as opções disponíveis, escolhemos uma para ser a **chave primária** da tabela e as outras opções restantes serão então denominadas como **chaves alternativas**

Aluno		
<u>matricula</u>	nome	cpf
2017001	José	111.222.333-44
2017002	Maria	777.888.999-00
2017003	João	555.666.777-88

**Chave primária** **Chave alternativa**

- Neste exemplo, tanto o campo `matricula` como o `cpf` poderão ser usados para distinguir um registro dos demais
- Mas a chave alternativa não poderá ser utilizada como uma chave estrangeira, apenas a chave primária

# ESTRUTURA DE UM SGBD RELACIONAL

## **Domínios** **e** **Valores Vazios (*Null*)**

# ESTRUTURA DE UM SGBD RELACIONAL

- **Domínios e Valores Vazios (Null)**
- Quando uma tabela é definida, devemos definir também qual o conjunto de valores que cada um dos seus campos poderão assumir
  - *Alfanumérico, numérico, booleano, data, hora* etc
- Ao conjunto de valores definidos para um campo (ou coluna) damos o nome de *domínio do campo* (ou *domínio da coluna*)
- Além de definir o domínio para o campo, devemos também definir **se ele poderá ou não receber "valores vazios"** (*null* em inglês)
  - Se um campo de um registro está **vazio** (*null*) significa que este campo **não recebeu nenhum valor do seu domínio**

# ESTRUTURA DE UM SGBD RELACIONAL

- **Domínios e Valores Vazios (Null)**
- Os campos onde **não são admitidos valores vazios** são chamados de **campos obrigatórios**
- Os campos **que aceitam valores vazios** são chamados de **campos opcionais**

The diagram shows a table named 'Funcionario' with four columns: 'codigo', 'nome', 'cod\_func\_gerente', and 'data\_nascimento'. The 'codigo' column is underlined. Annotations include: 'Não Aceitam NULL (obrigatórios)' pointing to 'codigo' and 'nome'; 'Aceitam NULL (opcionais)' pointing to 'cod\_func\_gerente' and 'data\_nascimento'. A 'campo vazio' label points to a greyed-out cell in the 'data\_nascimento' column of the second row.

<u>codigo</u>	nome	cod_func_gerente	data_nascimento
1	José		10/10/87
2	Maria	1	
3	Ana	1	31/01/78
4	João		13/02/73
5	Paula	4	
6	Antônio	4	25/03/45



# ESTRUTURA DE UM SGBD RELACIONAL

- **Domínios e Valores Vazios (Null)**
- Geralmente os SGBD's exigem que todos os campos da **chave primária** sejam obrigatórios (não vazios)
- Para as demais chaves, como a estrangeira, isso não é exigido.

**Funcionario**

<u>codigo</u>	nome	cod_func_gerente	data_nascimento
1	José		10/10/87
2	Maria	1	
3	Ana	1	31/01/78
4	João		13/02/73
5	Paula	4	
6	Antônio	4	25/03/45

**Não Aceitam NULL (obrigatórios)** (aponta para codigo)

**Aceitam NULL (opcionais)** (aponta para cod\_func\_gerente e data\_nascimento)

**campo vazio** (aponta para a célula vazia na linha 2, coluna 4)

# ESTRUTURA DE UM SGBD RELACIONAL

## Restrições de Integridade

# ESTRUTURA DE UM SGBD RELACIONAL

- **Restrições de Integridade**
- Já vimos que umas das vantagens de se utilizar um SGBD está no fato deste garantir automaticamente a **integridade dos dados**
  - *Integridade dos Dados* = *Dados Corretos e Consistentes*
- Um SGBD irá nos oferecer meios para definirmos quais **restrições de integridade** desejaremos impor aos dados armazenados

# ESTRUTURA DE UM SGBD RELACIONAL

- **Restrições de Integridade**
- As restrições de integridade são classificadas nas seguintes categorias:
  - **Integridade de Domínio:** o valor de um campo deverá obedecer à definição dos valores admitidos para ele (domínio de campo)
  - **Integridade de Vazio:** indica se o valor de um campo poderá ou não receber valores vazios (*null*)
  - **Integridade de Chave:** irá garantir que os valores da chave primária e da chave alternativa serão únicos para cada registro
  - **Integridade referencial:** irá garantir que os valores dos campos que são chaves estrangeiras deverão existir na chave primária da tabela referenciada

# ESTRUTURA DE UM SGBD RELACIONAL

## **Modelo Lógico de um Banco de Dados**

# ESTRUTURA DE UM SGBD RELACIONAL

- Finalizando, um **Modelo Lógico** que represente a **Especificação (ou Esquema) de um Banco de Dados Relacional** deverá conter no mínimo as seguintes definições:
  1. **Tabelas** que compõe o banco de dados
  2. **Campos** que estas tabelas possuem
  3. **Restrições de integridade**, sendo elas:
    - Integridade de Domínio (valores permitidos para os campos)
    - Integridade de Vazio (campos obrigatórios ou opcionais)
    - Integridade de Chave (primária e alternativa)
    - Integridade Referencial (chaves estrangeiras)



**FIM**