



**Universidade de Brasília**

Departamento de Ciência da Computação



# Bancos de Dados

CIC0097



**Prof. Pedro Garcia Freitas**

<https://pedrogarcia.gitlab.io/>

[pedro.garcia@unb.br](mailto:pedro.garcia@unb.br)

Universidade de Brasília  
Instituto de Ciências Exatas  
Departamento de Ciências da Computação



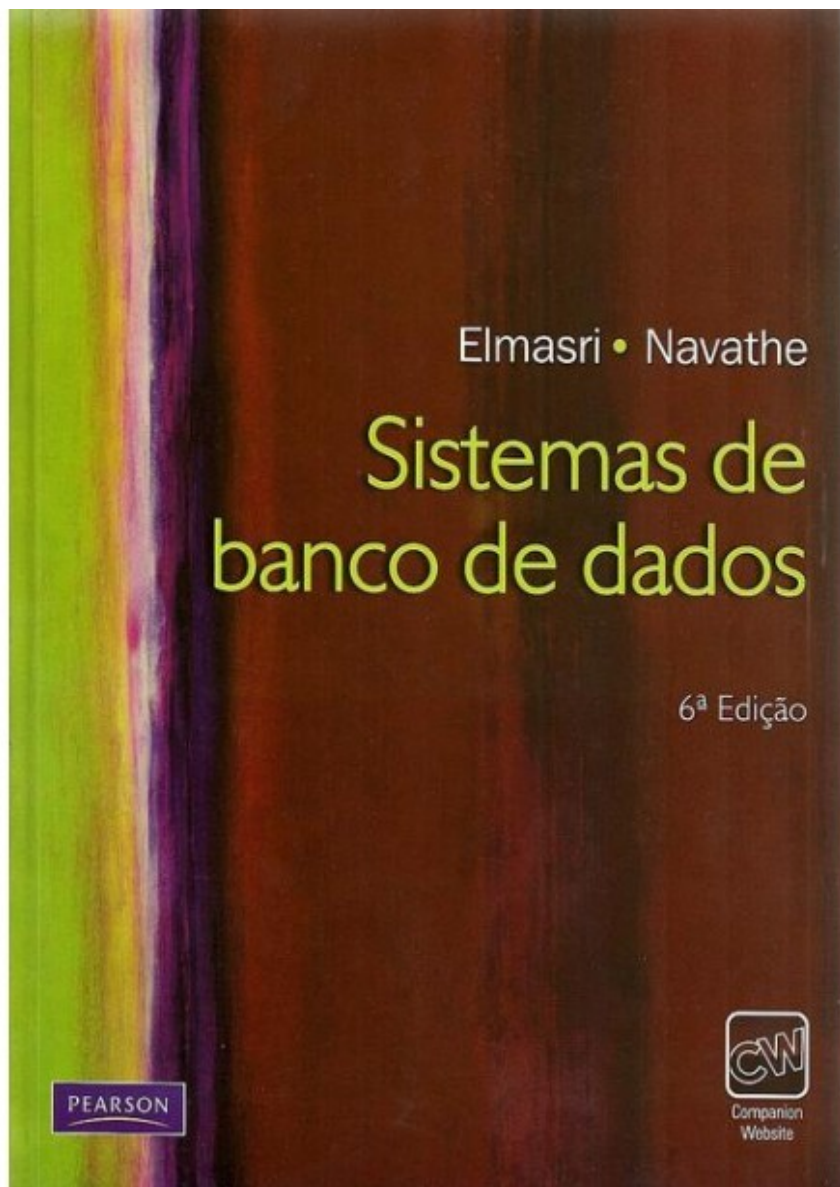
Este conjunto de slides não deve ser utilizado ou republicado sem a expressa permissão do autor.

This set of slides should not be used or republished without the author's express permission.



# **Projeto de banco de dados: dependências funcionais e normalização – PT1**

**CIC0097/2023.2  
T1/T2**



Esta aula se baseia  
no Capítulo 15 (SQL  
básica) do Elmasri e  
Navathe (6<sup>a</sup> Edição).



# Objetivos

Esta aula inicia uma nova fase da nossa disciplina que diz respeito à verificação de qualidade de projetos de banco de dados, começando pelas diretrizes informais para que tenhamos a possibilidade de criar um bom projeto de banco de dados.



# Projeto de Banco de Dados

- Até aqui, assumimos que os atributos são agrupados para formar um esquema de relação usando o bom senso do projetista de banco de dados ou mapeando um projeto de esquema de banco de dados com base no modelo de dados conceitual, como o MER.
- Porém, ainda precisamos de algum modo formal de análise porque um agrupamento de atributos em um esquema de relação pode ser melhor do que outro.



# Projeto de Banco de Dados

- Ao discutir o projeto de banco de dados nas aulas anteriores, não desenvolvemos nenhuma medida de adequação ou boas práticas para medir a qualidade do projeto, além da intuição do projetista.
- Nesta aula, vamos discutir parte da teoria que foi desenvolvida com o objetivo de avaliar esquemas relacionais para a qualidade do projeto.





# Projeto de Banco de Dados

Ou seja, para medir formalmente (ou informalmente) porque um conjunto de agrupamentos de atributos em esquemas de relação é melhor do que outro.





# Níveis

Existem dois níveis em que podemos discutir as boas práticas de esquemas de relação:

- Lógico (conceitual)
- Implementação (físico)

# Níveis

Existe

Consiste em avaliar como os usuários interpretam os esquemas de relação e o significado de seus atributos.

podemos discutir as boas práticas de esquemas de relação:

- **Lógico (conceitual)**
- Implementação (físico)



Nível

Exist

pod

de esquemas e relação:

- **Lógico (conceitual)**
- Implementação (físico)

Ter bons esquemas de relação nesse nível permite que os usuários entendam claramente o significado dos dados nas relações, e daí formulem suas consultas corretamente

# Níveis

Existência

persistência

de esquemas

de relação:

- Lógico (conceitual)

- **Implementação (físico)**

Avalia como as tuplas em uma relação da base são armazenadas e atualizadas.



Nível

Exis

pod

de

Esse nível se aplica apenas a esquemas das relações da base (fisicamente armazenadas como arquivos) enquanto no nível lógico estamos interessados em esquemas de relações da base e visões (relações virtuais).

- Logico (conceitual)

- Implementação (físico)



# Objetivos do Projeto de BD

- O projeto de banco de dados relacional por fim produz um conjunto de relações.
- Os objetivos implícitos da atividade de projeto são preservação da informação e redundância mínima.



# Diretrizes de projeto informais para esquemas de relação

Antes de discutirmos a *teoria formal* do projeto de banco de dados relacional, podemos considerar quatro *diretrizes informais* que podem ser usadas como medidas para determinar a qualidade de projeto.





# As diretrizes informais buscam:

- I. Garantir que a semântica dos atributos seja clara no esquema.
- II. Reduzir a informação redundante nas tuplas.
- III. Reduzir os valores NULL nas tuplas.
- IV. Eliminar a possibilidade de gerar tuplas falsas.



# I. Garantir que a semântica dos atributos seja clara no esquema

- O significado de uma relação deve estar fortemente associado à interpretação dos valores dos atributos em uma tupla;
- Em geral, quanto mais fácil for explicar a semântica da relação, melhor será o projeto do esquema de relação.



# I. Garantir que a semântica dos atributos seja clara no esquema

- Os esquemas deveriam minimizar a possibilidade de ambiguidades de interpretação;
- A facilidade em explicar o significado dos atributos em uma relação pode ser tomada como uma medida de qualidade de projeto.



## II. Reduzir a informação redundante nas tuplas

Um objetivo do projeto de esquema é minimizar o espaço de armazenamento usado pelas relações (e, portanto, pelos arquivos correspondentes). (**Economia de Espaço de Armazenamento**)



## II. Reduzir a informação redundante nas tuplas

A redundância aumenta o risco de inconsistências nos dados (manutenção de Consistência): Se uma informação redundante for atualizada em um lugar e não em outro, isso pode levar a discrepâncias e erros nos dados.



## II. Reduzir a informação redundante nas tuplas

**Eficiência da DML:** Sem redundância desnecessária, as operações de atualização e inserção tornam-se mais eficientes, pois você não precisa fazer alterações em vários locais para manter a consistência.



## II. Reduzir a informação redundante nas tuplas

**Eficiência da DQL:** Menos dados redundantes significam menos dados para serem processados, resultando em consultas mais rápidas e eficientes.





## II. Reduzir a informação redundante nas tuplas

### Evita Anomalias de Atualização:

Redundâncias podem levar a anomalias de atualização, como a atualização de parte dos dados, mas não em todos os lugares onde a informação está armazenada. Isso pode resultar em dados inconsistentes e imprecisos.



## II. Reduzir a informação redundante nas tuplas

**Manutenibilidade:** Reduzir a redundância facilita a modificação do esquema do banco de dados, tornando-o mais flexível para acomodar mudanças nos requisitos.

## II. Reduzir a informação redundante nas tuplas

**Promove a normalização:** Reduzir a redundância está intimamente ligado aos princípios de normalização do banco de dados.

- A normalização visa organizar os dados de maneira eficiente, minimizando a redundância e mantendo a integridade referencial.



## II. Reduzir a informação redundante nas tuplas

Em resumo, a redução da informação redundante contribui para um design de banco de dados mais eficiente, consistente e fácil de manter, resultando em melhor desempenho e qualidade dos dados.



## III. Reduzir os valores NULL das tuplas

- Se muitos dos atributos não se aplicarem a todas as tuplas na relação, acabamos com muitos NULLs nessas tuplas.
- Isso pode desperdiçar espaço no nível de armazenamento e também ocasionar problemas com o conhecimento do significado dos atributos e com a especificação de operações JUNÇÃO.



## III. Reduzir os valores NULL das tuplas

O uso excessivo de valores nulos pode levar a inconsistências nos dados. Se não for controlado adequadamente, pode resultar em informações imprecisas ou mal interpretadas.



## III. Reduzir os valores NULL das tuplas

Valores nulos podem complicar a manutenção da integridade referencial, especialmente em relações entre tabelas. Isso porque a correspondência entre chaves estrangeiras e primárias torna-se mais desafiadora quando há valores nulos.





## III. Reduzir os valores NULL das tuplas

A presença de valores nulos pode complicar a realização de operações e consultas, pois é necessário lidar com casos especiais. Isso aumenta a complexidade do código e pode afetar o desempenho das consultas.



## III. Reduzir os valores NULL das tuplas

**Promover Boas Práticas de Modelagem de Dados:** Minimizar valores nulos visa organizar os dados de maneira eficiente, reduzindo redundâncias e anomalias. Um design de banco de dados que minimize o uso de valores nulos tende a ser mais robusto, fácil de entender e manter.



## IV. Reduzir a possibilidade de gerar tuplas falsas

"Tuplas espúrias" (ou "tuplas falsas", Elmasri) geralmente refere-se a linhas ou registros que foram incluídos em uma tabela devido a operações de junção (join) que resultaram em combinações incorretas ou indesejadas de dados.



## IV. Reduzir a possibilidade de gerar tuplas falsas

São linhas resultantes de operações de junção que não correspondem adequadamente aos critérios de relacionamento entre as tabelas e, portanto, não representam relações válidas no contexto do banco de dados.



## IV. Reduzir a possibilidade de gerar tuplas falsas

Isso ocorre quando uma junção entre tabelas produz mais linhas do que o esperado devido a critérios de junção inadequados ou à falta de uma condição de junção apropriada.



## IV. Reduzir a possibilidade de gerar tuplas falsas

Para evitar tuplas falsas, é crucial usar as condições de junção corretas para garantir que apenas as combinações desejadas sejam incluídas no resultado. Condições de junção apropriadas geralmente envolvem a igualdade de chaves entre as tabelas.



EmpRoleProj

empName	role	projName
Smith	designer	Nile
Smith	programmer	Amazon
Smith	designer	Amazon
Jones	designer	Amazon

$\pi_{\text{role}, \text{projName}}$

Table 1

empName	role
Smith	designer
Smith	programmer
Jones	designer

$\pi_{\text{empName}, \text{role}}$

Table 2

role	projName
designer	Nile
programmer	Amazon
designer	Amazon





empName	role	projName	
Smith	designer	Nile	
Smith	designer	Amazon	
Smith	programmer	Amazon	
Jones	designer	Nile	← spurious tuple
Jones	designer	Amazon	

**SELECT \* From table1 JOIN table2 on  
table1.role=table2.role**

Table 1

empName	role
Smith	designer
Smith	programmer
Jones	designer

Table 2

role	projName
designer	Nile
programmer	Amazon
designer	Amazon



```
create table table1 (  
    empName varchar(100),  
    role varchar(100)  
);  
insert into table1 values ("smith", "designer");  
insert into table1 values ("smith", "programmer");  
insert into table1 values ("jones", "designer");  
  
create table table2 (  
    role varchar(100),  
    projName varchar(100)  
);  
  
insert into table2 values ("designer", "Nile");  
insert into table2 values ("programmer", "Amazon");  
insert into table2 values ("designer", "Amazon");  
  
SELECT * From table1 JOIN table2 on table1.role=table2.role;
```

# Diretrizes Informais

Objetivo: Garantir que a semântica dos atributos seja clara no esquema

**Primeira diretriz: Ao projetar esquemas de relação, garanta que seja simples explicar seu significado. Evite a combinação de atributos de diferentes tipos de entidade e tipos de relacionamento em uma única relação.**

# Diretrizes Informais

- Vejamos um exemplo de um projeto de banco de dados de baixa qualidade, **e que deve ser evitado.**

## Servidor\_Depto

<u>cpf</u>	s_nome	s_dt_nasc	endereco	depto_numero	depto_nome	depto_g_ident

## Servidor\_Projeto

<u>serv_cpf</u>	func_nome	horas	<u>proj_numero</u>	proj_nome	proj_localizacao



## 2. Projeto de Banco de Dados

- Anomalia da **inserção**

### Servidor\_Depto

<u>cpf</u>	s_nome	s_dt_nasc	endereco	depto_numero	depto_nome	depto_g_ident
1163	Claudia	12/08/1974	SP	5	Engenharia	1163
1164	Jorge	29/01/1986	SP	5	Engenharia	1163
1165	Moacir	08/11/1981	RS	4	Administração	1165

Como inserir um departamento que não possui **servidores**?

## 2. Projeto de Banco de Dados

- Anomalia da **inserção**

Servidor\_Depto

<u>cpf</u>	s_nome
1163	Claudio
1164	Jorge
1165	Moacir

A anomalia de inserção é uma situação que pode ocorrer quando há a possibilidade de inserir dados inconsistentes ou incompletos devido à estrutura do banco de dados.

Como

dent



## 2. Projeto de Banco de Dados

- Anomalia da inserção

### Servidor\_Projeto

<u>serv_cpf</u>	<u>serv_nome</u>	horas	<u>proj_numero</u>	proj_nome	proj_localizacao
1163	Claudia	4	1	ProdutoX	São Paulo
1164	Jorge	6	2	ProdutoY	Sorocaba
1165	Moacir	5	1	ProdutoX	São Paulo
1163	Claudia	6	3	ProdutoZ	São Paulo

**A inserção de novos projetos associados a um mesmo funcionário exige a repetição da informação sobre o funcionário.**



## 2. Projeto de Banco de Dados

- Anomalia da **inserção**

### Servidor\_Projeto

<u>serv_cpf</u>	<u>serv_nome</u>	horas	<u>proj_numero</u>	proj_nome	proj_localizacao
1163	Claudia	4	1	ProdutoX	São Paulo
1164	Jorge	6	2	ProdutoY	Sorocaba
1165	Moacir	5	1	ProdutoX	São Paulo
1163	Claudia	6	3	ProdutoZ	São Paulo

**A associação de mais de um funcionário a um projeto exige a repetição da informação sobre o projeto.**





## 2. Projeto de Banco de Dados

- Anomalia da exclusão

### Servidor\_Depto

cpf	s_nome	s_dt_nasc	endereco	depto_numero	depto_nome	depto_g_ident
1163	Claudia	12/08/1974	SP	5	Engenharia	1163
1164	Jorge	29/01/1986	SP	5	Engenharia	1163
1165	Moacir	08/11/1981	RS	4	Administração	1165

Se a tupla referente ao funcionário **Moacir** for excluída do sistema, as informações sobre o departamento de **Administração** deixarão de existir no sistema.



## 2. Projeto de Banco de Dados

- Anomalia da atualização

### Servidor\_Depto

cpf	s_nome	s_dt_nasc	endereco	depto_numero	depto_nome	depto_g_ident
1163	Claudia	12/08/1974	SP	5	Engenharia	1163
1164	Jorge	29/01/1986	SP	5	Engenharia	1163
1165	Moacir	08/11/1981	RS	4	Administração	1165

Se o nome do departamento 5 mudar, duas tuplas precisarão ser modificadas.

## 2. Projeto de Banco de Dados

- **Minimizar a redundância de informação nas tuplas das relações.**

**Segunda diretriz: ao projetar esquemas de relação, garanta que não seja possível ocorrer nenhuma anomalia durante as operações de inserção, exclusão ou modificação.**



## 2. Projeto de Banco de Dados

- Minimizar a presença de **valores nulos** nas tuplas:

**Terceira diretriz: Sempre que possível, evite criar atributos que recebam valores NULL frequentemente.**



## 2. Projeto de Banco de Dados

- Eliminar a chance de criar tuplas espúrias (falsas ou indesejadas).

**Quarta diretriz: Projetar esquemas de relação tal que elas possam ser combinadas em operações de junção com condição de igualdade em atributos que são pares relacionados apropriadamente (chave primária, chave estrangeira) sem que tuplas espúrias sejam geradas.**



### 3. Conclusão

- Resumidamente, temos 4 ***diretrizes informais*** que podem ser usadas como ***medidas para determinar a qualidade*** de projeto do esquema da relação:

1. Garantir que a semântica dos atributos seja clara no esquema.
2. Reduzir a informação redundante nas tuplas.
3. Reduzir os valores **NULL** nas tuplas.
4. Anular a possibilidade de gerar tuplas espúrias.

A word cloud featuring the word "THANK YOU" in large, bold, black capital letters. Surrounding it are various translations of "Thank You" in different languages, including: GRACIAS, ARIGATO, SHUKURIA, GOZAIMASHITA, EFCHARISTO, JUSPAXAR, DANKSCHEEN, TASHAKKUR ATU, YAQHANYELAY, SUKSAMA, EKHMET, BİYAN, SHUKRIA, TINGKI, GRAZIE, MEHRBANI, PALMES, BOLZİN, and MERCI. The words are arranged in a circular pattern around the central "THANK YOU" text.



# Dúvidas?

