

# BANCO DE DADOS RELACIONAL

**GROUP BY com JOINS e Relatórios Avançados**

***Professora:***

Lucineide Pimenta

# Recapitulando a aula anterior



- ✓ Na aula passada, trabalhamos com **funções de agregação, GROUP BY e HAVING**.  
Os exercícios foram:
  - ✓ Listar quantos livros cada autor possui (BD Biblioteca).
  - ✓ Mostrar a média de páginas dos livros por editora (BD Biblioteca).
  - ✓ Listar o total de campanhas por reservatório (limnologia\_db).
  - ✓ Mostrar a média de valores de cada parâmetro em séries temporais (limnologia\_db).
  - ✓ Mostrar a média de valores de cada parâmetro em séries temporais (limnologia\_db).

# BANCO DE DADOS RELACIONAL

Sistema de Biblioteca - Modelagem e Implementação do Banco de Dados

# Modelo Entidade-Relacionamento (MER)

- ❑ Entidades principais:
- ❑ **Autor**
  - ❑ id\_autor (PK)
  - ❑ nome
- ❑ **Livro**
  - ❑ id\_livro (PK)
  - ❑ titulo
  - ❑ ano\_publicacao
  - ❑ id\_autor (FK → Autor)
- ❑ **Aluno**
  - ❑ id\_aluno (PK)
  - ❑ nome
  - ❑ curso
- ❑ **Emprestimo**
  - ❑ id\_emprestimo (PK)
  - ❑ data\_emprestimo
  - ❑ id\_aluno (FK → Aluno)
- ❑ **EmprestimoLivro** (associativa N:M)
  - ❑ id\_emprestimo (FK → Emprestimo)
  - ❑ id\_livro (FK → Livro)
- ❑ **Assim temos:**
  - ❑ Relação 1:N entre Autor → Livro.
  - ❑ Relação 1:N entre Aluno → Emprestimo.
  - ❑ Relação N:M entre Emprestimo ↔ Livro (via EmprestimoLivro).

# Script de Criação — schema\_biblioteca.sql

-- Criar o banco

```
CREATE DATABASE biblioteca;
```

-- Conectar ao banco

```
\c biblioteca;
```

-- Tabela Autor

```
CREATE TABLE autor (
```

```
    id_autor SERIAL PRIMARY KEY,
```

```
    nome VARCHAR(100) NOT NULL
```

```
);
```

-- Tabela Livro

```
CREATE TABLE livro (
```

```
    id_livro SERIAL PRIMARY KEY,
```

```
    titulo VARCHAR(150) NOT NULL,
```

```
    ano_publicacao INT,
```

```
    id_autor INT REFERENCES autor(id_autor)
```

```
);
```

-- Tabela Aluno

```
CREATE TABLE aluno (
```

```
    id_aluno SERIAL PRIMARY KEY,
```

```
    nome VARCHAR(100) NOT NULL,
```

```
    curso VARCHAR(100) NOT NULL
```

```
);
```

# Script de Inserts — dados\_iniciais\_biblioteca.sql

*-- Autores*

```
INSERT INTO autor (nome) VALUES  
('J. R. R. Tolkien'),  
('Machado de Assis'),  
('Clarice Lispector');
```

*-- Livros*

```
INSERT INTO livro (titulo, ano_publicacao,  
id_autor) VALUES  
('O Senhor dos Anéis', 1954, 1),  
('Dom Casmurro', 1899, 2),  
('A Hora da Estrela', 1977, 3),  
('O Hobbit', 1937, 1);
```

*-- Alunos*

```
INSERT INTO aluno (nome, curso) VALUES  
('Ana Souza', 'Sistemas de Informação'),  
('Bruno Silva', 'Engenharia de Software');
```

*-- Empréstimos*

```
INSERT INTO emprestimo (data_emprestimo,  
id_aluno) VALUES  
('2025-08-20', 1),  
('2025-08-21', 2);
```

# Script de Inserts — dados\_iniciais\_biblioteca.sql

*-- EmprestimoLivro (associativa)*

*INSERT INTO emprestimo\_livro (id\_emprestimo, id\_livro) VALUES*

*(1, 1), -- Ana Souza pegou O Senhor dos Anéis*

*(1, 2), -- Ana Souza pegou Dom Casmurro*

*(2, 3); -- Bruno Silva pegou A Hora da Estrela*

# Correção (exemplos de soluções)

- ❑ Exercício 1 — Listar quantos livros cada autor possui (**BD Biblioteca**)

```
SELECT a.nome AS autor, COUNT(l.id_livro) AS total_livros
```

```
FROM autor a
```

```
INNER JOIN livro l ON a.id_autor = l.id_autor
```

```
GROUP BY a.nome
```

```
ORDER BY total_livros DESC;
```



# Correção (exemplos de soluções)

- ❑ Exercício 2 — Mostrar a média de páginas dos livros por editora (**BD Biblioteca**)

```
SELECT e.nome AS editora, AVG(l.num_paginas) AS media_paginas  
FROM editora e  
INNER JOIN livro l ON e.id_editora = l.id_editora  
GROUP BY e.nome  
ORDER BY media_paginas DESC;
```

# Correção (exemplos de soluções)

- ❑ Exercício 3 — Listar o total de campanhas por reservatório (**limnologia\_db**)

```
SELECT r.nome AS reservatorio, COUNT(c.id_campanha) AS total_campanhas  
FROM reservatorio r  
INNER JOIN campanha c ON r.id_reservatorio = c.id_reservatorio  
GROUP BY r.nome  
ORDER BY total_campanhas DESC;
```

# Correção (exemplos de soluções)

- ❑ Exercício 4 — Mostrar a média de valores de cada parâmetro em séries temporais (**limnologia\_db**)

```
SELECT p.nome_parametro, AVG(s.valor) AS media_valores  
FROM serie_temporal s  
INNER JOIN parametro p ON s.id_parametro = p.id_parametro  
GROUP BY p.nome_parametro  
ORDER BY media_valores DESC;
```

# Correção (exemplos de soluções)

- ❑ Exercício 5 — Exibir apenas as instituições que realizaram mais de 3 campanhas (**limnologia\_db**)

```
SELECT i.nome AS instituicao, COUNT(c.id_campanha) AS total_campanhas
FROM instituicao i
INNER JOIN campanha c ON i.id_instituicao = c.id_instituicao
GROUP BY i.nome
HAVING COUNT(c.id_campanha) > 3
ORDER BY total_campanhas DESC;
```

# Correção (exemplos de soluções)

- ❑ Exercício 5 — Exibir apenas as instituições que realizaram mais de 3 campanhas (**limnologia\_db**)

```
SELECT i.nome AS instituicao, COUNT(c.id_campanha) AS total_campanhas
FROM instituicao i
INNER JOIN campanha c ON i.id_instituicao = c.id_instituicao
GROUP BY i.nome
HAVING COUNT(c.id_campanha) > 3
ORDER BY total_campanhas DESC;
```

# BANCO DE DADOS RELACIONAL

## GROUP BY com múltiplos JOINS

# Objetivos da aula



- ✓ Revisão rápida de funções agregadoras.
- ✓ Uso de **GROUP BY com múltiplos JOINS**.
- ✓ Relatórios combinando várias tabelas.
- ✓ Filtragem com **HAVING em consultas complexas**.
- ✓ Exemplos práticos no **BD Biblioteca** e no **limnologia\_db**.

# GROUP BY em consultas com múltiplas tabelas

- ❑ A biblioteca quer saber **quantos livros cada autor publicou em cada editora**. Isso ajuda a identificar se um autor concentra suas publicações em uma única editora ou diversifica.

*-- Contar quantos livros cada autor possui por editora*

*SELECT a.nome AS autor, e.nome AS editora, COUNT(l.id\_livro) AS total\_livros*

*FROM autor a*

*INNER JOIN livro l ON a.id\_autor = l.id\_autor*

*INNER JOIN editora e ON l.id\_editora = e.id\_editora*

*GROUP BY a.nome, e.nome*

*ORDER BY total\_livros DESC;*



# GROUP BY em consultas com múltiplas tabelas

- ❑ O INPE quer saber **quantas campanhas foram realizadas em cada reservatório, por cada instituição**. Esse relatório mostra a participação das instituições em cada local de coleta.

-- Total de campanhas por reservatório e instituição

```
SELECT r.nome AS reservatorio, i.nome AS instituicao, COUNT(c.id_campanha) AS  
total_campanhas
```

```
FROM campanha c
```

```
INNER JOIN reservatorio r ON c.id_reservatorio = r.id_reservatorio
```

```
INNER JOIN instituicao i ON c.id_instituicao = i.id_instituicao
```

```
GROUP BY r.nome, i.nome
```

```
ORDER BY total_campanhas DESC;
```

# Usando HAVING em consultas complexas

- ❑ A biblioteca quer identificar **editoras que publicaram mais de 5 livros**. Isso permite ver quais editoras são mais relevantes para o acervo.

*-- Mostrar apenas editoras que publicaram mais de 5 livros*

*SELECT e.nome AS editora, COUNT(l.id\_livro) AS total\_livros*

*FROM editora e*

*INNER JOIN livro l ON e.id\_editora = l.id\_editora*

*GROUP BY e.nome*

*HAVING COUNT(l.id\_livro) > 5;*

# Usando HAVING em consultas complexas

- ❑ O INPE precisa saber **quais reservatórios tiveram mais de 10 campanhas de coleta**. Isso permite priorizar os locais mais monitorados.

*-- Mostrar apenas reservatórios que tiveram mais de 10 campanhas*

*SELECT r.nome AS reservatorio, COUNT(c.id\_campanha) AS total\_campanhas*

*FROM reservatorio r*

*INNER JOIN campanha c ON r.id\_reservatorio = c.id\_reservatorio*

*GROUP BY r.nome*

*HAVING COUNT(c.id\_campanha) > 10;*

# Relatórios úteis no contexto do INPE

- ❑ O INPE quer verificar a **média dos valores de cada parâmetro em cada reservatório** (ex.: temperatura, oxigênio dissolvido).

```
SELECT r.nome AS reservatorio, p.nome_parametro, AVG(s.valor) AS media_valores  
FROM serie_temporal s  
INNER JOIN reservatorio r ON s.id_reservatorio = r.id_reservatorio  
INNER JOIN parametro p ON s.id_parametro = p.id_parametro  
GROUP BY r.nome, p.nome_parametro;
```

# Relatórios úteis no contexto do INPE

- ❑ O INPE precisa identificar **instituições que coletaram em mais de um reservatório**, para avaliar a abrangência de cada parceiro.

```
SELECT i.nome AS instituicao, COUNT(DISTINCT c.id_reservatorio) AS  
total_reservatorios  
FROM instituicao i  
INNER JOIN campanha c ON i.id_instituicao = c.id_instituicao  
GROUP BY i.nome  
HAVING COUNT(DISTINCT c.id_reservatorio) > 1;
```

# Atividade Prática (Individual)

- ❑ Criar o arquivo:

*/scripts/exercicios\_aula12.sql*

- ❑ **Exercícios:**

- ❑ Listar quantos livros cada autor publicou por editora (**BD Biblioteca**).
  - ❑ Listar a média de páginas dos livros por autor (**BD Biblioteca**).
  - ❑ Mostrar o total de campanhas por reservatório e instituição (**limnologia\_db**).
  - ❑ Mostrar a média de valores de parâmetros por reservatório (**limnologia\_db**).
  - ❑ Listar as instituições que coletaram em mais de um reservatório (**limnologia\_db**).
- 
- ❑ Cada consulta deve estar comentada no script, explicando o que retorna.

# Encerramento

- ❑ **Nesta aula aprendemos:**
  - ❑ Como usar **GROUP BY** em consultas com múltiplos JOINS.
  - ❑ Como aplicar **HAVING** em relatórios complexos.
  - ❑ Como gerar relatórios úteis para o INPE com dados agrupados.

# O que veremos na próxima aula

- ❑ Consolidação de consultas com **subconsultas + GROUP BY**.
- ❑ Como combinar **subqueries** com agregações.
- ❑ Preparação final para o **Requisito BDR.02 — Funções Agrupadoras**.





# Referências Bibliográfica da Aula

## Livros:

**Elmasri & Navathe (2010).** Sistemas de Banco de Dados.

**Silberschatz et al. (2011).** Sistemas de Banco de Dados.

## Links úteis:

 [PostgreSQL Docs](#)

 [DBDiagram.io](#)

# Bibliografia Básica

- ❑ DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro, Elsevier: Campus, 2004.
- ❑ ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 7 ed. São Paulo: Pearson, 2018.
- ❑ SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. F. **Sistema de banco de dados**. Rio de Janeiro: Elsevier Brasil, 2016.

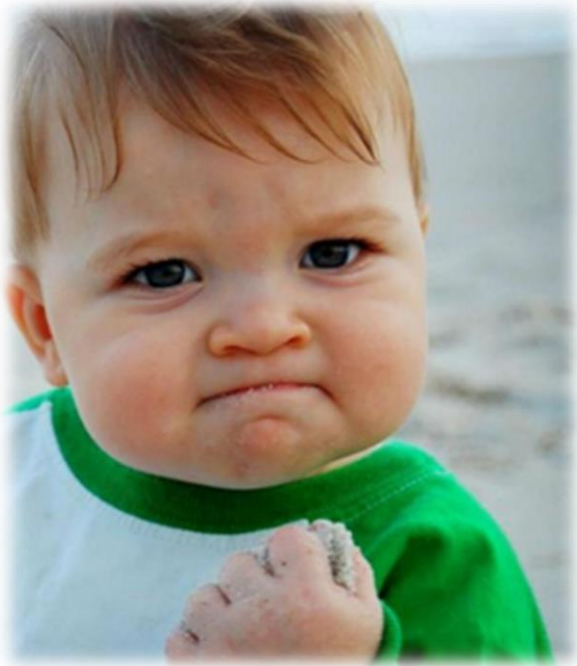
# Bibliografia Complementar

- ❑ BEAULIEU, A. **Aprendendo SQL**. São Paulo: Novatec, 2010.
- ❑ GILLENSON, M. L. **Fundamentos de Sistemas de Gerência de Banco de Dados**. Rio de Janeiro: LTC, 2006.
- ❑ MACHADO, F. N. R. **Banco de Dados: Projeto e Implementação**. São Paulo: Érica, 2005.
- ❑ OTEY, M; OTEY, D. **Microsoft SQL Server 2005: Guia do Desenvolvedor**. Rio de Janeiro: Ciência Moderna, 2007.
- ❑ RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de Gerenciamento de Bancos de Dados**. 3 ed. Porto Alegre: Bookman, 2008.
- ❑ ROB, P; CORONEL, C. **Sistemas de Banco de Dados: Projeto, Implementação e Gerenciamento**. 8 ed. São Paulo: Cengage Learning, 2011.
- ❑ TEOREY, T; LIGHTSTONE, S; NADEAU, T. **Projeto e Modelagem de Bancos de Dados**. São Paulo: Campus, 2006.

# Dúvidas?



# Considerações Finais



**Professora:  
Lucineide Pimenta**

**Bom descanso à todos!**

