



# INSTITUTO SUPERIOR DO LITORAL DO PARANÁ

**Prof. Luiz Efigênio**

**Administração de Banco de Dados II  
Visão Geral e Conceitos Avançados de SQL**

## **Sobre o Professor**

### **Luiz Carlos Efigênio**

27 anos – Desenvolvedor

Graduado em Análise e Desenvolvimento de Sistemas – IFPR Paranaguá

Pós-graduado em Desenvolvimento FullStack – Descomplica

### **Experiências Profissionais:**

- Embarca – Backend Developer Engineer
- Omnichat – Backend Developer Engineer
- Omnichat – DevOps Engineer
- TopInformation – DevOps Engineer
- Algar Tech – Analista de Suporte

GitHub: [github.com/efigenioluiz](https://github.com/efigenioluiz)

LinkedIn: [linkedin.com/in/efigenioluiztads](https://www.linkedin.com/in/efigenioluiztads)

---

## **1. Introdução à Disciplina**

Nesta disciplina, os estudantes irão aprofundar os conhecimentos adquiridos em Banco de Dados I, explorando operações SQL avançadas, segurança da informação em bancos, otimização de consultas e introdução a bancos de dados não relacionais. O objetivo é proporcionar uma base sólida para administração eficiente de bancos em cenários reais.

### **Objetivos da disciplina:**

- Aprofundar comandos SQL para manipulação e análise de dados.
- Aplicar técnicas de segurança em sistemas gerenciadores de banco de dados (SGBDs).
- Compreender o funcionamento de transações e controle de concorrência.
- Conhecer fundamentos de bancos NoSQL orientados a documentos.

**INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá**

Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil



## 2. Extração, Filtro e Junção de Dados

### Conceitos abordados:

- Comando **SELECT** e cláusulas de filtragem (**WHERE, AND, OR, BETWEEN, IN, LIKE**).
- Tipos de junções:
  - **INNER JOIN**: combina registros comuns entre tabelas.
  - **LEFT JOIN**: mantém todos os registros da tabela da esquerda.
  - **RIGHT JOIN**: mantém todos os registros da tabela da direita.
  - **FULL OUTER JOIN**: une todos os registros, com ou sem correspondência.

### Exemplo prático:

```
SELECT c.nome, p.valor
```

```
FROM clientes c
```

```
INNER JOIN pedidos p ON c.id = p.cliente_id
```

```
WHERE p.valor > 100;
```

### Atividade sugerida:

- Criação de duas tabelas relacionadas (ex: **clientes** e **pedidos**).
- Consultas de extração com filtros e junções entre elas.

---

## 3. Agrupamento e Agregação de Dados

### Conceitos abordados:

- Comando **GROUP BY** para agrupamento de resultados.
- Funções agregadas: **SUM, AVG, COUNT, MAX, MIN**.
- Cláusula **HAVING** para filtragem após agrupamento.

### Exemplo:

```
SELECT cliente_id, COUNT(*) AS total_pedidos
```

```
FROM pedidos
```

```
GROUP BY cliente_id
```

```
HAVING COUNT(*) > 3;
```

**INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá**

Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil



#### Atividade sugerida:

- Gerar relatórios de vendas por cliente ou região.
- 

## 4. Funções de Transformação

### 4.1 Funções Aritméticas

- **ROUND(valor, casas\_decimais)** – Arredondamento.
- **MOD(dividendo, divisor)** – Resto da divisão.

### 4.2 Funções de Texto

- **LOWER()** / **UPPER()** – Converter texto.
- **SUBSTRING(campo, início, tamanho)** – Extrair parte da string.
- **LENGTH()** – Contar caracteres.

#### Exemplo:

```
SELECT UPPER(nome), ROUND(salario, 2) FROM funcionarios;
```

---

## 5. Subconsultas (Subqueries)

#### Conceitos abordados:

- Subconsultas simples e correlacionadas.
- Utilização em SELECT, FROM e WHERE.

#### Exemplo:

```
SELECT nome FROM produtos  
WHERE preco > (SELECT AVG(preco) FROM produtos);
```

#### Atividade:

- Listar clientes com valores acima da média dos pedidos.
- 

## 6. Visões (Views)

**INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá**  
Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil

**Conceito:**

- Uma view é uma tabela virtual baseada em uma query. Útil para abstração e segurança.

**Comando:**

```
CREATE VIEW vw_pedidos_resumo AS
```

```
SELECT cliente_id, COUNT(*) AS total
```

```
FROM pedidos
```

```
GROUP BY cliente_id;
```

---

## 7. Transações e Controle de Concorrência

**Conceito ACID:**

- **Atomicidade, Consistência, Isolamento, Durabilidade.**

**Comandos:**

- **BEGIN TRANSACTION, COMMIT, ROLLBACK.**

**Problemas comuns:** dirty read, phantom read, non-repeatable read.

**Exercício simulado:**

- Criar atualização de saldo com rollback em caso de erro.
- 

## 8. Segurança em Bancos SQL

**Tópicos:**

- Criação e gerenciamento de usuários.
- Privilégios com **GRANT** e **REVOKE**.
- Boas práticas: uso de roles, criptografia de dados sensíveis, controle de acesso por IP.

**Exemplo:**

**INSTITUTO SUPERIOR DO LITORAL DO PARANÁ | Paranaguá**

Rua João Eugênio, 534 - Costeira, Paranaguá - PR | 83203-400 – Brasil



GRANT SELECT ON pedidos TO user\_consulta;

---

## 9. Performance em Bancos SQL

### Técnicas:

- Criação de índices para acelerar consultas.
- Leitura de planos de execução (**EXPLAIN**).
- Avaliar quando usar desnormalização.

### Atividade:

- Criar índice em uma tabela grande e comparar tempo de resposta com e sem ele.
- 

## 10. Bancos de Dados Orientados a Documentos

### Conceitos principais:

- Armazenamento baseado em documentos JSON.
- Mais flexível que modelo relacional.

### Exemplo: MongoDB

```
{  
  "nome": "João",  
  "pedidos": [ { "data": "2023-01-01", "valor": 100 } ]  
}
```

---

## Encerramento da Aula 1

- Apresentação do cronograma das 4 aulas.
  - Alinhamento de expectativas da disciplina.
  - Dúvidas iniciais dos alunos.
  - Atividade diagnóstica: revisão SQL com SELECT, WHERE e JOIN.
-



## Referências Bibliográficas

- MACHADO, Felipe Nery Rodrigues. *Banco de dados: projeto e implementação*. 4. ed., São Paulo, Saraiva, 2020. (Digital).
- SANTOS, Marcela Gonçalves dos. *Algoritmos e programação*. 1. ed., Porto Alegre, SAGAH, 2017. (Digital).
- WAZLAWICK, Raul. *Introdução a algoritmos e programação com Python: uma abordagem dirigida por testes*. 1. ed., Rio de Janeiro, LTC, 2017. (Digital).