



# INSTITUTO SUPERIOR DO LITORAL DO PARANÁ

Prof. Luiz Efigênio

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

## Filtros (WHERE e Operadores)

O cláusula **WHERE** é utilizada para restringir os resultados de uma consulta. Ela atua sobre **linhas individuais da tabela** antes de qualquer agrupamento ou agregação.

Segundo **Elmasri & Navathe (2011, p. 135)**, o uso de filtros é essencial para recuperar apenas os dados necessários, melhorando tanto a performance da consulta quanto a clareza dos resultados.

### Principais operadores:

- Comparação: `=, <>, >, <, >=, <=`
- Intervalo: `BETWEEN ... AND ...`
- Conjunto: `IN ( ... )`
- Padrão: `LIKE` (usa `%` para qualquer sequência e `_` para um único caractere)
- Lógico: `AND, OR, NOT`

### 📌 Exemplo

```
SELECT *  
FROM Funcionarios  
WHERE Salario BETWEEN 3000 AND 5000  
AND Departamento = 'TI';
```

## 2. Junções Múltiplas (JOINS)

As **junções (JOINS)** permitem combinar dados de tabelas diferentes, estabelecendo

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



relacionamentos entre elas.

De acordo com **Silberschatz, Korth & Sudarshan (2012, p. 151)**, a operação de junção é uma das mais importantes da álgebra relacional, pois reflete diretamente os relacionamentos existentes no modelo entidade-relacionamento.

### Tipos principais:

- **INNER JOIN** → retorna apenas os registros que possuem correspondência em ambas as tabelas.
- **LEFT JOIN** → retorna todos os registros da tabela da esquerda e os correspondentes da direita (se existirem).
- **RIGHT JOIN** → retorna todos os registros da tabela da direita e os correspondentes da esquerda.
- **FULL JOIN** → retorna todos os registros de ambas as tabelas (quando suportado pelo SGBD).

#### 📌 Exemplo (junção tripla):

```
SELECT f.Nome, d.NomeDepartamento, p.NomeProjeto  
FROM Funcionarios f  
INNER JOIN Departamentos d ON f.DepartamentoID = d.ID  
INNER JOIN Projetos p ON f.ProjetoID = p.ID;
```

---

## 3. Funções de Agregação

As funções de agregação permitem realizar cálculos sobre conjuntos de registros.

Segundo **Elmasri & Navathe (2011, p. 141)**, essas funções são fundamentais para análises estatísticas e sumarizações em bancos de dados.

### Principais funções:

- **COUNT(\*)** → quantidade de registros
- **SUM(coluna)** → soma dos valores
- **AVG(coluna)** → média
- **MAX(coluna)** → maior valor



- **MIN(coluna)** → menor valor

💡 Exemplo:

```
SELECT AVG(Salario) AS MediaSalarial,  
       MAX(Salario) AS MaiorSalario  
FROM Funcionarios;
```

## 4. Agrupamento de Resultados (GROUP BY)

O **GROUP BY** é utilizado quando queremos **agrupar registros** com base em valores iguais em uma ou mais colunas.

De acordo com **Silberschatz, Korth & Sudarshan (2012, p. 161)**, a cláusula GROUP BY é essencial para relatórios que exigem agregações por categorias, como departamentos ou regiões.

💡 Exemplo:

```
SELECT Departamento, AVG(Salario) AS MediaSalarial  
FROM Funcionarios  
GROUP BY Departamento;
```



## 5. Filtrando Grupos (HAVING)

A cláusula **HAVING** é semelhante ao WHERE, mas atua sobre **grupos já formados pelo GROUP BY**, filtrando agregações.

Segundo **Elmasri & Navathe (2011, p. 144)**, HAVING foi introduzido no SQL justamente porque o WHERE não pode ser aplicado diretamente sobre funções de agregação.

📌 Exemplo:

```
SELECT Departamento, AVG(Salario) AS MediaSalarial  
FROM Funcionarios  
GROUP BY Departamento  
HAVING AVG(Salario) > 4000;
```

---

## Atividades Propostas

1. Liste todos os funcionários do departamento de "Vendas" com salário acima de 2500.
2. Exiba o nome dos funcionários, junto com o nome do departamento e do projeto em que trabalham.
3. Calcule o salário médio de cada departamento.
4. Mostre quais departamentos possuem salário médio maior que 5000.
5. Liste a quantidade de funcionários em cada projeto.



## Referências

- **SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.** *Sistemas de Banco de Dados*. 6ª Edição. São Paulo: Pearson, 2012.
- **ELMASRI, Ramez; NAVATHE, Shamkant B.** *Sistemas de Banco de Dados*. 7ª Edição. São Paulo: Pearson, 2011.
- Documentação oficial do **MySQL**: <https://dev.mysql.com/doc/>
- Documentação oficial do **PostgreSQL**: <https://www.postgresql.org/docs/>