



BD: JOINS, Subconsultas, Ordenação e Agrupamento

IFBA – Campus Ilhéus
Curso: Back End com Python
Prof.^a: Karine Prado

Objetivo da Aula

- ▶ **Objetivo:** Integrar dados de múltiplas tabelas.



INNER JOIN

- ▶ Usado para combinar registros de duas ou mais tabelas com base em uma coluna comum entre elas.
- ▶ Retorna apenas os registros com correspondência nas duas tabelas.
- ▶ Exemplo:

```
SELECT *  
FROM Aluno  
INNER JOIN Curso ON Aluno.CursoID =  
Curso.CursoID;
```

LEFT JOIN

- ▶ Retorna todos os registros da tabela da esquerda, e os correspondentes da tabela da direita.
- ▶ Os que não têm correspondência na direita vêm como **NULL**.
- ▶ Exemplo:

```
SELECT *
FROM Aluno
LEFT JOIN Curso ON Aluno.CursoID = Curso.CursoID;
```

RIGHT JOIN

- ▶ Retorna todos os registros da tabela da direita, e os correspondentes da esquerda.
- ▶ Os que não têm correspondência na esquerda vêm como **NULL**.
- ▶ Exemplo:

```
SELECT *  
FROM Aluno  
RIGHT JOIN Curso ON Aluno.CursoID =  
Curso.CursoID;
```

Subconsultas (Subqueries)

- ▶ É uma consulta dentro de outra consulta, usada para:
 - Filtrar dados
 - Realizar cálculos
 - Comparar valores com base em resultados de outra tabela

Subconsulta

- ▶ Exemplo:

```
SELECT Nome  
FROM Aluno  
WHERE CursoID = (  
    SELECT CursoID  
    FROM Curso  
    WHERE Nome = 'Python'  
);
```

ORDER BY

- ▶ Ordena os resultados por uma ou mais colunas.
- ▶ Padrão: ASC (crescente), use DESC para decrescente.
- ▶ Exemplo:

```
SELECT Nome, Nota  
FROM Aluno  
ORDER BY Nota DESC;
```

GROUP BY

- ▶ Agrupa os dados por uma ou mais colunas.
- ▶ Usado com funções de agregação: COUNT, SUM, AVG, etc.
- ▶ Exemplo:

```
SELECT CursoID, COUNT(*) AS TotalAlunos  
FROM Aluno  
GROUP BY CursoID;
```

HAVING

- ▶ Filtra grupos após a agregação (diferente do WHERE, que filtra linhas).
- ▶ Exemplo:

```
SELECT CursoID, COUNT(*) AS TotalAlunos  
FROM Aluno  
GROUP BY CursoID  
HAVING COUNT(*) > 10;
```