



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

Desenvolvimento WEB II

Banco de Dados e SQL: SQL e Joins.



Roteiro

- Banco de Dados e SQL: SQL e Joins:
 - Criação de Tabelas e Inserção de Dados;
 - Introdução ao JOIN;
 - Tipos de JOINS;
 - Exemplos Práticos de JOINS;
 - Desempenho e Otimização;
 - Atividade.



Criação de Tabelas e Inserção de Dados

Criação de Tabelas:

```
1  CREATE TABLE alunos (  
2      id INT PRIMARY KEY AUTO_INCREMENT,  
3      nome VARCHAR(100)  
4  );  
5  
6  CREATE TABLE cursos (  
7      id INT PRIMARY KEY AUTO_INCREMENT,  
8      nome VARCHAR(100),  
9      aluno_id INT,  
10     FOREIGN KEY (aluno_id) REFERENCES alunos(id)  
11 );
```



Criação de Tabelas e Inserção de Dados

Inserção de Dados:

```
1  INSERT INTO alunos (nome)
2  VALUES ('Ana'), ('Bruno'), ('Carlos'), ('Diana');
3
4  INSERT INTO cursos (nome, aluno_id) VALUES
5  ('Matemática', 1),
6  ('Física', 2),
7  ('Química', NULL),
8  ('Biologia', 3);
```



Criação de Tabelas e Inserção de Dados

Dados Iniciais:

Tabela “alunos”:

| id | nome |
|----|--------|
| 1 | Ana |
| 2 | Bruno |
| 3 | Carlos |
| 4 | Diana |



Criação de Tabelas e Inserção de Dados

Dados Iniciais:

Tabela “cursos”:

| id | nome | aluno_id |
|----|------------|----------|
| 1 | Matemática | 1 |
| 2 | Física | 2 |
| 3 | Química | NULL |
| 4 | Biologia | 3 |



Introdução ao JOIN

JOIN é uma operação em SQL que permite combinar registros de duas ou mais tabelas com base em uma condição relacionada entre elas.



Introdução ao JOIN

Sintaxe Básica:

```
1  SELECT colunas
2  FROM tabela1
3  JOIN tabela2
4  ON tabela1.coluna_comum = tabela2.coluna_comum;
```



Tipos de JOINS

| Tipo de JOIN | Descrição |
|-------------------|--|
| INNER JOIN | Retorna registros que têm valores correspondentes em ambas as tabelas. |
| LEFT JOIN | Retorna todos os registros da tabela à esquerda e os registros correspondentes da tabela à direita. Os registros sem correspondência terão valores NULL. |
| RIGHT JOIN | Retorna todos os registros da tabela à direita e os registros correspondentes da tabela à esquerda. Os registros sem correspondência terão valores NULL. |
| FULL JOIN | Retorna registros quando há uma correspondência em uma das tabelas. Combina os resultados de LEFT JOIN e RIGHT JOIN. |
| CROSS JOIN | Retorna o produto cartesiano de duas tabelas. |



Exemplos Práticos de JOINS

INNER JOIN:

```
1  SELECT alunos.nome AS aluno, cursos.nome AS curso
2  FROM alunos
3  INNER JOIN cursos ON alunos.id = cursos.aluno_id;
```



Exemplos Práticos de JOINS

Resultado:

Tabela resultante após o **INNER JOIN**:

| aluno | curso |
|--------|------------|
| Ana | Matemática |
| Bruno | Física |
| Carlos | Biologia |



Exemplos Práticos de JOINS

LEFT JOIN:

```
1  SELECT alunos.nome AS aluno, cursos.nome AS curso
2  FROM alunos
3  LEFT JOIN cursos ON alunos.id = cursos.aluno_id;
```



Exemplos Práticos de JOINS

Resultado:

Tabela resultante após o **LEFT JOIN**:

| aluno | curso |
|--------|------------|
| Ana | Matemática |
| Bruno | Física |
| Carlos | Biologia |
| Diana | NULL |



Exemplos Práticos de JOINS

RIGHT JOIN:

```
1  SELECT alunos.nome AS aluno, cursos.nome AS curso
2  FROM alunos
3  RIGHT JOIN cursos ON alunos.id = cursos.aluno_id;
```



Exemplos Práticos de JOINS

Resultado:

Tabela resultante após o **RIGHT JOIN**:

| aluno | curso |
|--------|------------|
| Ana | Matemática |
| Bruno | Física |
| NULL | Química |
| Carlos | Biologia |



Desempenho e Otimização

- **Impacto dos JOINS na Performance:**
 - JOINS podem aumentar o tempo de execução das consultas devido ao processamento adicional necessário para combinar tabelas.
 - **O uso de índices** nas colunas envolvidas nos JOINS pode melhorar significativamente o desempenho.



Desempenho e Otimização

- **Boas Práticas:**

- Evitar JOINS desnecessários.
- Usar índices nas colunas de **chave estrangeira.**
- Simplificar consultas complexas utilizando **subconsultas ou tabelas temporárias.**



Atividade Prática

Tarefa:

1. Criar um banco de dados com tabelas apresentadas nos exemplos anteriores.
2. Insira os dados de exemplo nas tabelas correspondentes apresentadas nos exemplos anteriores.
3. Insira novos dados da sua escolha nessas tabelas (personalização);
4. Teste consultas SQL utilizando diferentes tipos de JOINS (por exemplo: teste as consultas apresentadas nos exemplos anteriores e outras na sua escolha).



Obrigado!

Questões?

