**Aula: Comando UPDATE no MySQL**

**Objetivos:**

1. Compreender o propósito e a sintaxe do comando `UPDATE`.
2. Aprender a aplicar o comando `UPDATE` para modificar dados em tabelas.
3. Utilizar cláusulas adicionais para refinar e restringir as atualizações.

---

## 1. Introdução ao Comando UPDATE

### 1.1 O que é o Comando `UPDATE`

- **Descrição:** O comando `UPDATE` é utilizado para modificar registros existentes em uma tabela no banco de dados.

**Sintaxe Básica:**
sql
Copiar código

```
UPDATE nome_da_tabela
SET coluna1 = valor1, coluna2 = valor2, ...
WHERE condição;
```

- 
- **Notas Importantes:**
  - O comando `UPDATE` sem uma cláusula `WHERE` atualizará todos os registros na tabela.
  - A cláusula `WHERE` é essencial para garantir que apenas os registros desejados sejam modificados.

---

## 2. Sintaxe e Estrutura do Comando UPDATE

### 2.1 Sintaxe Básica

- **Exemplo Simples:**
  - Suponha uma tabela `funcionarios` com as colunas `id`, `nome`, e `salario`.

sql
Copiar código

```
UPDATE funcionarios
SET salario = 5000
WHERE id = 1;
```

- 

    - Neste exemplo, apenas o funcionário com `id` igual a 1 terá seu salário atualizado para 5000.

**2.2 Atualizar Múltiplas Colunas**

- **Exemplo:**
    - Para atualizar o salário e o nome de um funcionário específico:

sql
Copiar código
```
UPDATE funcionarios
SET salario = 5500, nome = 'João Silva'
WHERE id = 2;
```

- 

**2.3 Atualizar Vários Registros**

- **Exemplo:**
    - Suponha que queremos aumentar o salário de todos os funcionários no departamento 'TI' em 10%:

sql
Copiar código
```
UPDATE funcionarios
SET salario = salario * 1.10
WHERE departamento = 'TI';
```

- 

**2.4 Atualizar com Subconsulta**

- **Exemplo:**
    - Atualize o salário dos funcionários com base em uma tabela de ajustes de salário:

sql
Copiar código
```
UPDATE funcionarios
SET salario = (SELECT novo_salario FROM ajustes_salario WHERE
ajustes_salario.id = funcionarios.id)
WHERE EXISTS (SELECT 1 FROM ajustes_salario WHERE ajustes_salario.id
= funcionarios.id);
```

- 

---

## 3. Cuidados e Boas Práticas

### 3.1 Utilização da Cláusula WHERE

- **Importância da Cláusula WHERE:**
  - Sempre use a cláusula `WHERE` para especificar quais registros devem ser atualizados.
  - Sem a cláusula `WHERE`, o comando atualizará todos os registros na tabela, o que pode causar perda de dados.

### 3.2 Revisão Antes da Atualização

- **Passo a Passo:**
  - Antes de executar uma atualização, faça uma consulta `SELECT` para revisar os registros que serão alterados.

sql
Copiar código
```
SELECT * FROM funcionarios WHERE departamento = 'TI';
```

- 

### 3.3 Backup de Dados

- **Recomendação:**
  - Sempre faça backup dos dados antes de executar operações de atualização massiva para evitar perda acidental de dados.

---

## 4. Exemplos de Aplicação Prática

### 4.1 Exemplo 1: Atualização Simples

Atualize o nome do funcionário com `id` 3 para 'Maria Oliveira':
sql
Copiar código
```
UPDATE funcionarios
SET nome = 'Maria Oliveira'
WHERE id = 3;
```

- 

### 4.2 Exemplo 2: Atualização Condicional

Aumente o salário em 5% para todos os funcionários cujo salário atual seja menor que 4000:
sql

Copiar código

```
UPDATE funcionarios
SET salario = salario * 1.05
WHERE salario < 4000;
```

- 

#### 4.3 Exemplo 3: Atualização com Subconsulta

Atualize o status de pedidos para 'Entregue' onde o `id` do pedido está na lista de pedidos entregues:
sql
Copiar código

```
UPDATE pedidos
SET status = 'Entregue'
WHERE id IN (SELECT id FROM pedidos_entregues);
```

- 

---

## 5. Exercícios Práticos

1. **Atualizar Dados Específicos:**
   - Atualize o salário de um funcionário com `id` específico para um novo valor.
2. **Atualização Condicional:**
   - Aumente o salário de todos os funcionários em um departamento específico.
3. **Atualização com Subconsulta:**
   - Atualize um campo na tabela `clientes` com base nos dados de uma outra tabela `novos_dados_clientes`.
4. **Atualização em Massa:**
   - Aplique um aumento percentual no salário para todos os funcionários com mais de 5 anos de empresa.