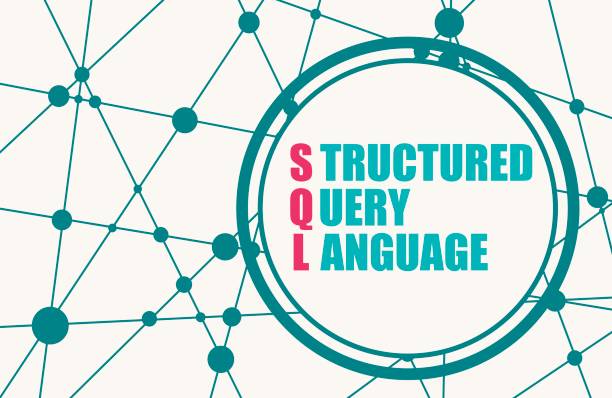
**SQL: cláusulas UPDATE e DELETE**



## Introdução: SQL — cláusulas UPDATE e DELETE

Após a finalização de um projeto de modelagem de dados e a sua implementação, inicia-se o desenvolvimento para a manutenção do banco de dados. Essa manutenção se dá por meio do [SGBD](https://www.alura.com.br/artigos/sgbds-relacionais), que nos permite realizar vários processos nos bancos de dados, como **consultar, inserir, alterar e excluir** dados das tabelas.

Para alterar e excluir dados de uma tabela, utilizamos duas cláusulas bem importantes da **linguagem SQL**: a cláusula UPDATE que é responsável por alterar os dados armazenados e a cláusula DELETE que é responsável por remover os dados.

Neste artigo, vamos conhecer a sintaxe básica de cada cláusula e entender quais os riscos e os cuidados necessários ao executar as cláusulas UPDATE e DELETE.

## Cláusula UPDATE



Utilizamos a cláusula UPDATE para realizar alterações nos dados armazenados em uma tabela do banco de dados. A sintaxe básica do comando utilizado por todos os bancos de dados relacionais é:

UPDATE nome\_da\_tabela

SET coluna = valor,

WHERE condição;

Na cláusula UPDATE, informamos o nome da tabela que queremos atualizar, utilizamos o SET para indicar os campos da tabela que serão atualizados e no WHERE expressamos a condição para a atualização, ou seja, especificamos quais registros devem ser atualizados na tabela.

## Cláusula DELETE



Utilizamos a cláusula DELETE para realizar a exclusão de dados de uma ou mais tabelas de um banco de dados. A sintaxe básica do comando utilizado por todos os bancos de dados relacionais é:

DELETE

FROM nome\_da\_tabela,

WHERE condição;

Onde na cláusula FROM, informamos o nome da tabela que queremos excluir os dados, e no WHERE informamos a condição que especifica quais registros devem ser excluídos da tabela.

## Riscos e Cuidados ao executar os comandos

Ao executar os comandos para realizar a atualização ou exclusão de dados de uma tabela de um banco de dados, precisamos tomar alguns cuidados.

Nesse sentido, um ponto de atenção é sempre informar uma condição ao realizar uma atualização ou exclusão de dados de uma tabela. Quando não informamos essa condição, corremos o risco de que todos os dados da tabela sejam atualizados, ou até mesmo excluídos, ocorrendo assim a perda de dados importantes.

## Cláusula WHERE

Para que essa perda de dados em massa não ocorra, utilizamos a cláusula ´WHERE´. Especificamos nesta cláusula os critérios que os dados armazenados em uma tabela devem cumprir para que os registros que contêm esses parâmetros sejam incluídos nos resultados da consulta.

Para aplicarmos na prática a utilização das cláusulas DELETE e 'UPDATE'. Vamos observar o seguinte exemplo:

Em um banco de dados existe a tabela de clientes e a tabela de vendedores, com os seguintes campos e registros:

| **ID** | **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | 1471156710 | Érica Carvalho | R. Iriquitia | Jardins | São Paulo | SP | 80012212 |
| 02 | 19290992743 | Fernando Cavalcante | R. Dois de Fevereiro | Agua Santa | Rio de Janeiro | RJ | 22000000 |
| 03 | 2600586709 | César Teixeira | Rua Conde de Bonfim | Tijuca | Rio de Janeiro | RJ | 22020001 |
| 04 | 492472718 | Eduardo Jorge | R. Volta Grande | Tijuca | Rio de Janeiro | RJ | 22012002 |
| 05 | 50534475787 | Abel Silva | Rua Humaitá | Humaitá | Rio de Janeiro | RJ | 22000212 |
| 06 | 5576228758 | Petra Oliveira | R. Benício de Abreu | Lapa | São Paulo | SP | 88192029 |
| 07 | 5840119709 | Gabriel Araujo | R. Manuel de Oliveira | Santo Amaro | São Paulo | SP | 80010221 |

| **MATRÍCULA** | **NOME** | **BAIRRO** | **COMISSÃO** | **DATA ADMISSÃO** |
| --- | --- | --- | --- | --- |
| 235 | Márcio Almeida | Tijuca | 0.08 | 2014-08-15 |
| 236 | Cláudia Morais | Jardins | 0.08 | 2013-09-17 |
| 237 | Roberta Martins | Copacabana | 0.11 | 2017-03-18 |
| 238 | Péricles Alves | Santo Amaro | 0.11 | 2016-08-21 |

Estas duas tabelas, serão utilizadas em todos os exemplos apresentados durante o artigo.

## UPDATE Com WHERE

Tendo então a tabela de clientes acima como nosso exemplo, precisamos atualizar as informações da cliente que possui o CPF `147115670´ e está na primeira linha da nossa tabela, para isso vamos utilizar a seguinte consulta:

UPDATE CLIENTES

SET nome = 'Érica Silvia'

WHERE CPF = '1471156710';

Ao executar este comando apenas a cliente que possui o CPF `147115670´, terá os seus dados alterados:

| **ID** | **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | **1471156710** | **Érica Silva** | R. Iriquitia | Jardins | São Paulo | SP | 80012212 |
| 02 | 19290992743 | Fernando Cavalcante | R. Dois de Fevereiro | Agua Santa | Rio de Janeiro | RJ | 22000000 |
| 03 | 2600586709 | César Teixeira | Rua Conde de Bonfim | Tijuca | Rio de Janeiro | RJ | 22020001 |
| 04 | 492472718 | Eduardo Jorge | R. Volta Grande | Tijuca | Rio de Janeiro | RJ | 22012002 |
| 05 | 50534475787 | Abel Silva | Rua Humaitá | Humaitá | Rio de Janeiro | RJ | 22000212 |
| 06 | 5576228758 | Petra Oliveira | R. Benício de Abreu | Lapa | São Paulo | SP | 88192029 |
| 07 | 5840119709 | Gabriel Araujo | R. Manuel de Oliveira | Santo Amaro | São Paulo | SP | 80010221 |

Para atualizar mais de um campo ao mesmo tempo, podemos utilizar a seguinte consulta:

UPDATE CLIENTES

SET NOME = 'Fernando Sousa',CEP = '80012212'

WHERE CPF = '19290992743';

Assim, apenas as informações de nome e CEP do segundo cliente que possui o CPF 19290992743 será atualizado:

| **ID** | **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | 1471156710 | Érica Silva | R. Iriquitia | Jardins | São Paulo | SP | 80012212 |
| 02 | 19290992743 | **Fernando Sousa** | R. Dois de Fevereiro | Agua Santa | Rio de Janeiro | RJ | **80012212** |
| 03 | 2600586709 | César Teixeira | Rua Conde de Bonfim | Tijuca | Rio de Janeiro | RJ | 22020001 |
| 04 | 492472718 | Eduardo Jorge | R. Volta Grande | Tijuca | Rio de Janeiro | RJ | 22012002 |
| 05 | 50534475787 | Abel Silva | Rua Humaitá | Humaitá | Rio de Janeiro | RJ | 22000212 |
| 06 | 5576228758 | Petra Oliveira | R. Benício de Abreu | Lapa | São Paulo | SP | 88192029 |
| 07 | 5840119709 | Gabriel Araujo | R. Manuel de Oliveira | Santo Amaro | São Paulo | SP | 80010221 |

## UPDATE sem WHERE

Precisamos agora alterar os dados de um outro registro armazenado na nossa tabela de clientes:

UPDATE Clientes

SET ESTADO = 'SP';

Como a cláusula WHERE não foi utilizada, informando o registro que deveria ser atualizado, o valor do campo**ESTADO** de todos clientes foram alterados:

| **ID** | **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | 1471156710 | Érica Silva | R. Iriquitia | Jardins | São Paulo | **SP** | 80012212 |
| 02 | 19290992743 | Fernando Sousa | R. Dois de Fevereiro | Agua Santa | Rio de Janeiro | **SP** | 80012212 |
| 03 | 2600586709 | César Teixeira | Rua Conde de Bonfim | Tijuca | Rio de Janeiro | **SP** | 22020001 |
| 04 | 492472718 | Eduardo Jorge | R. Volta Grande | Tijuca | Rio de Janeiro | **SP** | 22012002 |
| 05 | 50534475787 | Abel Silva | Rua Humaitá | Humaitá | Rio de Janeiro | **SP** | 22000212 |
| 06 | 5576228758 | Petra Oliveira | R. Benício de Abreu | Lapa | São Paulo | **SP** | 88192029 |
| 07 | 5840119709 | Gabriel Araujo | R. Manuel de Oliveira | Santo Amaro | São Paulo | **SP** | 80010221 |

## DELETE Com WHERE

Um dos clientes, solicitou que as suas informações fossem removidas do banco de dados. Para realizar esta exclusão, vamos utilizar a seguinte consulta:

DELETE

FROM Clientes

WHERE CPF = '5840119709';

Ao executar este comando apenas o cliente **Gabriel Araujo** que estava localizado na ultima linha da tabela de clientes, foi excluído:

| **ID** | **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | 1471156710 | Érica Silva | R. Iriquitia | Jardins | São Paulo | **SP** | 80012212 |
| 02 | 19290992743 | Fernando Sousa | R. Dois de Fevereiro | Agua Santa | Rio de Janeiro | SP | 80012212 |
| 03 | 2600586709 | César Teixeira | Rua Conde de Bonfim | Tijuca | Rio de Janeiro | SP | 22020001 |
| 04 | 492472718 | Eduardo Jorge | R. Volta Grande | Tijuca | Rio de Janeiro | SP | 22012002 |
| 05 | 50534475787 | Abel Silva | Rua Humaitá | Humaitá | Rio de Janeiro | SP | 22000212 |
| 06 | 5576228758 | Petra Oliveira | R. Benício de Abreu | Lapa | São Paulo | SP | 88192029 |

## DELETE sem WHERE

Pensando que se a cláusula WHERE não fosse utilizada ao executar um comando DELETE para remover algum registro, teríamos como resultado a exclusão de todos os dados armazenados na tabela, ou seja, perdendo os dados de todos os clientes:

DELETE

FROM Clientes;

| **CPF** | **NOME** | **ENDEREÇO** | **BAIRRO** | **CIDADE** | **ESTADO** | **CEP** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Atualizar utilizando como condição outras tabelas

Podemos informar na cláusula WHERE como condição para realizar atualização ou exclusão dos dados, outra consulta, que pode ser utilizada para buscar informações armazenadas na própria tabela ou em **outras tabelas**:

Sendo assim, vamos analisar o exemplo abaixo:

UPDATE VENDEDORES

SET COMISSÃO = COMISSÃO + 0.03

WHERE COMISSÃO = (SELECT min(COMISSÃO) FROM VENDEDORES)

O SELECT passado na cláusula WHERE, retornará apenas o valor da menor comissão armazenada na tabela. Dessa forma, apenas os vendedores que possuem o valor da comissão igual ao valor retornado neste SELECT terão os seus dados alterados e o valor da sua comissão aumentará:

| **MATRÍCULA** | **NOME** | **BAIRRO** | **COMISSÃO** | **DATA ADMISSÃO** |
| --- | --- | --- | --- | --- |
| 235 | Márcio Almeida | Tijuca | **0.11** | 2014-08-15 |
| 236 | Cláudia Morais | Jardins | **0.11** | 2013-09-17 |
| 237 | Roberta Martins | Copacabana | 0.11 | 2017-03-18 |
| 238 | Péricles Alves | Santo Amaro | 0.11 | 2016-08-21 |

## Chave primária

Uma outra forma de garantir que apenas os dados desejados sofram alterações é a criação de uma chave primária na tabela.

**A chave primária, ou Primary key (PK) é o dado que pode ser utilizado como um identificador único de um registro em uma tabela no banco de dados.**

Se definirmos que o campo CPF será a chave primária da tabela de clientes, estamos definindo que este campo receberá apenas valores únicos. Ao utilizarmos este campo como uma condição no momento de executar uma consulta com as cláusulas DELETE ou UPDATE, garantimos que apenas o registro que possui aquele dado será alterado.

## Conclusão

Como estudamos neste artigo, as cláusulas DELETE e UPDATE são muito importantes na manipulação de um banco de dados, pois, através delas, conseguimos atualizar ou excluir dados de uma tabela.

Também entendemos os riscos que corremos ao utilizar estas cláusulas e como podemos evitar que, ao executar comandos de modo não recomendado, ocorram perdas de dados em massa em um banco de dados.

Portanto, podemos utilizar estas cláusulas em diversas consultas, assim como utilizar outros recursos disponibilizados pela linguagem SQL para garantir que não haja perda de dados.