

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

SQL

Comandos utilizados na manipulação de bancos de dados

Quando você utiliza algum aplicativo e faz alguma pesquisa, seja qual for, no celular, sites de músicas, filmes, etc., lembre-se que você estará acessando um banco de dados. Este banco de dados foi manipulado para inserir, atualizar ou deletar informações e após uma recuperação de dados estes são exibidos a você. Para essa manipulação é necessário utilizar o principal subconjunto de instruções SQL, a linguagem de manipulação de dados (DML), e as instruções que irão inserir (INSERT), atualizar ou modificar dados (UPDATE), assim como excluir dados (DELETE) em tabelas.

A cláusula SELECT

O resultado de uma consulta SQL é uma tabela. Na sintaxe da instrução SELECT temos a cláusula WHERE, os conectivos AND, OR e NOT podem também ser utilizados. Também pode-se utilizar operadores de comparação <, <=, >, >=, = e <>.

Eles podem ser utilizados com *strings* e expressões aritméticas além de tipos especiais e tipos de data. A cláusula FROM especifica uma ou mais tabelas de sua consulta ([MACHADO, 2014](#)).

Exemplos:

Exibição de toda a tabela

```
SELECT * FROM convidado;
```

Exibição de todos os convidados com o nome iniciando com 'A'

```
SELECT * FROM convidado WHERE nome LIKE 'A%';
```

Instrução de Inserção (INSERT)

A instrução INSERT permite adicionar novas linhas, ou registros numa tabela existente. Sua sintaxe é:

```
INSERT  
[INTO] nome_tabela  
[(nome_coluna [, nome_coluna] ...)]  
{VALUES | VALUE} (lista_valores) [, (lista_valores)] ...
```

Declaração VALUES

A declaração VALUES especificará os valores ou lista de valores explícitos para serem inseridos. Caso a instrução seja utilizada sem parâmetros:

```
INSERT INTO nome_tabela () VALUES();
```

Veja exemplos de algumas instruções:

Criação da tabela

```
CREATE TABLE IF NOT EXISTS convidado (  
  id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(50) NOT NULL DEFAULT '',  
  nascimento DATE,  
  estudante ENUM('Não', 'Sim') NOT NULL DEFAULT 'Não'  
);
```

Inserção de valores

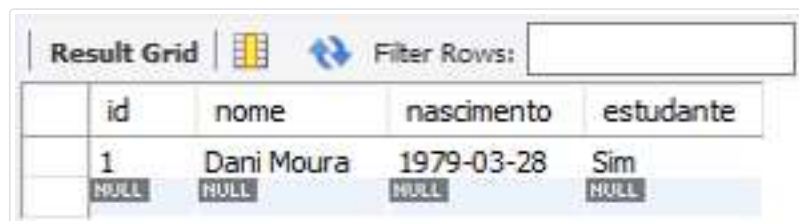
```
INSERT INTO CONVIDADO (nome, nascimento, estudante)  
VALUES ('Dani Moura', '1979-03-28', 'Sim');
```

A instrução acima insere uma linha à tabela “convidado”, incluindo nas colunas “nome”, “nascimento” e “estudante” os valores respectivos da cláusula VALUES. Não é necessário utilizar a ordem de criação das colunas na instrução.

Visualização da consulta

Para verificar o resultado, utilize a sintaxe a seguir:

```
SELECT * FROM convidado;
```



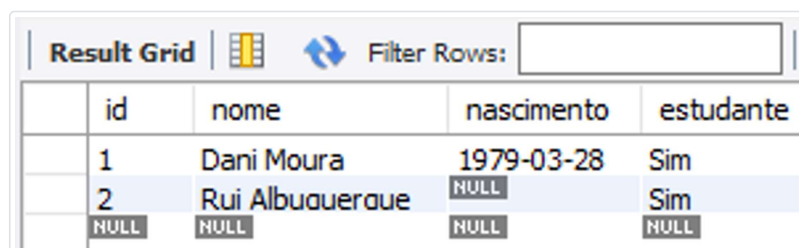
	id	nome	nascimento	estudante
	1	Dani Moura	1979-03-28	Sim
	NULL	NULL	NULL	NULL

Fonte: elaborada pelo autor, captura de tela software MySQL WorkBench.

Atribuição de valor nulo

Quando a coluna não for especificada, nada será atribuído e, ao consultar esta coluna será exibido como NULL (nulo). Para atribuir explicitamente um valor nulo a instrução deverá ser:

```
INSERT INTO CONVIDADO (nome, nascimento, estudante)  
VALUES ('Rui Albuquerque', null , 'Sim');
```



	id	nome	nascimento	estudante
	1	Dani Moura	1979-03-28	Sim
	2	Rui Albuquerque	NULL	Sim
	NULL	NULL	NULL	NULL

Fonte: elaborada pelo autor, captura de tela software MySQL WorkBench.

Atenção: Fique atento, pois se a coluna “nascimento” fosse criada com a opção NOT NULL, esta instrução não seria executada, pois esta é uma restrição que exige um valor para a coluna.

Instrução de Atualização (UPDATE)

UPDATE é uma instrução DML que altera ou atualiza linhas em uma tabela. Quando se deseja definir expressões de tabelas comuns acessíveis no UPDATE, esta instrução pode começar com uma cláusula WITH. Veja o exemplo desta sintaxe a seguir:

```
UPDATE tabela_referência
SET lista_atribuição
[WHERE condição]
[ORDER BY ...]
[LIMIT quantidade_linhas]
value:
{expr | DEFAULT}
assignment:
nome_coluna = valor
lista_atribuições:
atribuição [, atribuição] ...
```

[Saiba Mais](#)

Exemplo do comando UPDATE

```
UPDATE convidado
SET estudante = 'Sim'
WHERE nome = 'Lebrencio Grulher'
AND nascimento = '08-Jul-1990';
```

Explicação do exemplo

Este comando atualiza como sendo estudantes todos os convidados de nome Lebrencio Grulher que tenham nascido na data de 08 de Julho de 1990. Homônimos de Lebrencio Grulher que tenham nascido em datas diferentes, ou outros convidados que não se chamam Lebrencio Grulher mas que tenham nascido em 08 de Julho de 1990 não serão atualizados para a condição de estudante.

Cláusula WHERE

Se fornecida, a cláusula WHERE, especifica as condições que identificam quais linhas devem ser atualizadas. Fique atento que se não for apresentada nenhuma cláusula WHERE, todas as linhas serão atualizadas.

Por exemplo, para determinar que na tabela “convidado” a coluna “estudante” seja atualizada para “Não”, como na sintaxe a seguir, esta atualização ocorrerá em todas as linhas da tabela pela falta da cláusula WHERE.

```
UPDATE convidado SET estudante = 'Não';
```

Cláusulas ORDER BY e LIMIT

Se a cláusula ORDER BY for especificada, as linhas serão atualizadas na ordem especificada. A cláusula LIMIT coloca um limite no número de linhas que podem ser atualizadas. A instrução a seguir exemplifica a cláusula ORDER BY:

```
UPDATE convidado
SET estudante = 'Sim'
WHERE nascimento < '08-Jul-1990'
ORDER BY nome;
```

Explicação do exemplo ▾

Este comando vai atualizar todos os cadastrados nascidos antes de 08 de Julho de 1990 como sendo estudantes.

Quando adicionamos um limite por meio da cláusula `LIMIT`, o que teremos é uma limitação nas atualizações. No exemplo anterior, se quisermos limitar o número de atualização aos primeiros 10 nomes (em ordem alfabética), teremos:

```
UPDATE convidado
  SET estudante = 'Sim'
 WHERE nascimento < '08-Jul-1990'
  LIMIT 10
 ORDER BY nome;
```

Instrução de Exclusão (DELETE)

DELETE é uma instrução DML que exclui linhas de uma tabela. Sua sintaxe geral é dada por:

```
DELETE FROM nome_tabela
 [WHERE condição]
 [ORDER BY ...]
 [LIMIT quantidade_linhas]
```

Assim como para a instrução UPDATE, as condições na cláusula WHERE irão identificar quais linhas serão excluídas. E sem a utilização da cláusula WHERE, todas as linhas são excluídas. Quando declarada, a condição é avaliada a cada linha, e se verdadeira ela é excluída. Exemplo:

```
DELETE FROM convidados
  WHERE estudante = 'Sim'
  ORDER BY nome
  LIMIT 10;
```

Explicação do exemplo ▾

Este comando deleta os dez primeiros estudantes (por ordem alfabética) classificados como estudantes.

- **ORDER BY** - Se a cláusula ORDER BY estiver presente, as linhas serão excluídas na ordem especificada.
- **LIMIT** - A cláusula LIMIT coloca um limite no número de linhas que podem ser excluídas. Mesmo quando você excluir a linha que contém o valor máximo de uma coluna "AUTO_INCREMENT", o valor não será reutilizado.
- **DELETE** - Se a instrução DELETE incluir uma cláusula ORDER BY, as linhas serão excluídas na ordem especificada pela cláusula.

Quando utilizadas em conjunto, as cláusulas ORDER BY e LIMIT são bastante úteis. Veja a sintaxe a seguir:

```
DELETE FROM log_usuario WHERE usuario = 'rm'
 ORDER BY datahora_acao LIMIT 1;
```

Explicação do exemplo ▾

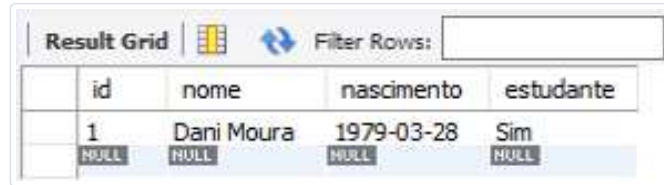
A instrução localiza linhas correspondentes ao que foi especificado na cláusula WHERE, classificando-as pela coluna "datahora_acao" do tipo DATETIME e exclui a primeira, ou seja, a mais antiga.

Aplicando esta instrução ao nosso banco de dados, mais especificamente em a tabela convidado, teremos:

```
DELETE FROM convidado WHERE id = 2
```

Após a instrução o resultado da consulta a tabela “convidado” será conforme a imagem a seguir, com o convidado 2 deletado:

Visualização da consulta da tabela convidado



The screenshot shows the MySQL Workbench Result Grid interface. At the top, there's a 'Result Grid' tab, a grid icon, a refresh icon, and a 'Filter Rows:' input field. Below this is a table with four columns: 'id', 'nome', 'nascimento', and 'estudante'. The first row contains the values '1', 'Dani Moura', '1979-03-28', and 'Sim'. The second row contains 'NULL', 'NULL', 'NULL', and 'NULL'.

id	nome	nascimento	estudante
1	Dani Moura	1979-03-28	Sim
NULL	NULL	NULL	NULL

Fonte: elaborada pelo autor, captura de tela software MySQL WorkBench.



Fonte: Shutterstock.

A manipulação de banco de dados, inclui as funções básicas de inserção, deleção e atualização, para podermos fazer a manutenção de dados em qualquer banco de dados. Isso significa que precisamos utilizar as instruções do subconjunto DML (Linguagem de manipulação de dados), elas são as principais instruções e as mais utilizadas. Aqui, você viu alguns exemplos de como utilizar as instruções de inserção (INSERT), de atualização (UPDATE) e, ou exclusão de dados (DELETE).