



BANCOS DE DADOS

Profº Me. Jones Artur Gonçalves

INSERINDO DADOS A TABELA

O comando para inclusão no banco de dados é o **INSERT**, que possui a seguinte estrutura:

```
INSERT INTO nome_tabela (lista-de-campos) VALUES  
(lista_dados)
```

Ou

```
INSERT INTO nome_tabela VALUES (lista_dados)
```

INSERT

➤ Sintaxe

Nome_tabela: nome da tabela no qual será inserido os dados.

Lista-de-campos: nome das colunas que receberão os valores.

Lista-dados: valores que serão inseridos na tabela.

Estes campos devem estar na mesma ordem descrita em lista-de-campos, todos separados por vírgula.

INSERT

➤ Syntaxe

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Ou

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

INSERT

➤ Sintaxe

A declaração SQL abaixo ilustra a inclusão de uma única linha de dados na tabela ESTADO E CIDADE.

```
Insert into estado  
values  
('SP', 'São Paulo');
```

```
Insert into cidade  
values  
('1', 'Votorantim', 'SP');
```

Não sendo especificadas as colunas no comando insert, os valores da cláusula values devem respeitar a ordem em que foram criados os campos

```
Insert into cidade (Codcidade, Nomecidade, Cdestado)  
Values ('2', 'Sorocaba', 'SP');
```

INSERT

➤ Sintaxe

A declaração SQL abaixo ilustra a inclusão de uma única linha de dados na tabela ESTADO E CIDADE.

```
Insert into cidade (Codcidade, Nomecidade, Cdestado)  
Values ('3', 'Itu', 'NN');
```

```
Insert into cidade (Codcidade, Nomecidade, Cdestado)  
Values ('4', 'Itapira', 'NN');
```

```
Insert into estado  
values  
('NN', 'Não Cadastrado');
```

Atualizando Dados

Em certas situações, podemos desejar mudar um valor em uma tupla sem mudar *todos* os valores na tupla. Para isto, a instrução `UPDATE` pode ser usada.

UPDATE

➤ Sintaxe

Atualiza os dados de um registro ou um grupo de registros de uma tabela do banco de dados

Sintaxe:

Update <nome-tabela>

Set <nome-coluna> = <novo conteúdo para o campo>

[<nome-coluna> = <novo conteúdo para o campo>]

Where <condição>

Opera somente em uma tabela .

SET - especifica as colunas a serem alteradas e os valores novos.

WHERE - especifica quais as linhas que serão atualizadas.

Se os novos valores violarem as restrições de integridade o comando UPDATE não é efetivado.

UPDATE

➤ Sintaxe

Update <nome-tabela>

Set <nome-coluna> = <novo conteúdo para o campo>

[<nome-coluna> = <novo conteúdo para o campo>]

Where <condição>

Onde

- a) nome-coluna representa a coluna da tabela que terá os dados modificados
- b) novo conteúdo para o campo representa o novo valor a ser atribuído ao campo
- c) condição representa a condição para a seleção dos registros que serão atualizados. Esta seleção poderá resultar em um ou vários registros.

UPDATE

➤ Sintaxe

Exemplo:

```
Update Cidade  
Set Cdestado= 'SP'  
Where Cdestado='NN'
```

O banco de dados retorna a mensagem:

(2 row(s) affected)

Excluindo dados

Remove linhas de uma tabela.

Sintaxe:

```
DELETE FROM <nome_tabela>  
WHERE <condição_de_seleção>;
```

Podemos **remover** apenas tuplas inteiras ;

Não podemos **remover** valores apenas em atributos particulares.

DELETE

➤ Exemplo

Apagar todos os vendedores com faixa de comissão nula

```
DELETE FROM cidade  
WHERE nomecidade = 'Itapira'
```

A cláusula DELETE remove uma ou mais linhas de uma única tabela , de acordo com a condição especificada na cláusula WHERE.

Se a cláusula WHERE não for especificada todas as linhas serão removidas.

DELETE

➤ Exemplo

Remover da tabela de pedidos os pedidos que contêm produtos que custem menos de R\$0,20

Versão ANSI

```
DELETE FROM pedido
WHERE num_pedido IN
  (SELECT num_pedido
   FROM produto P, item_pedido I
   WHERE P.cod_produto = I.cod_produto and
        valor_unitario < $0.20)
```

DELETE

➤ Exemplo

Remover da tabela de pedidos os pedidos que contenham produtos que custem menos de R\$0,20

Versão Transaction SQL

```
DELETE FROM pedido
FROM produto P, item_pedido I
WHERE I.num_pedido = pedido.num_pedido AND
P.cod_produto = I.cod_produto AND
valor_unitario < $0.20
```

Auto Incremento

A propriedade **IDENTITY** é utilizada para atributos (campos/colunas) das tabelas nas funções **CREATE TABLE**, e tem como finalidade incrementar um valor a cada nova inserção.

A sintaxe para usar esta propriedade é: **IDENTITY [(início , incremento)]**

Onde:

- **Início:** Valor a ser utilizado para o primeiro valor inserido na coluna.
- **Incremento:** Valor a ser incrementado a cada nova inserção.

Auto Incremento

Exemplo para criação de uma tabela utilizando um campo auto incremento:

```
CREATE TABLE Tbl_Clientes(  
    Codigo INT IDENTITY(1,1) PRIMARY KEY,  
    Nome VARCHAR(100),  
    UF VARCHAR(2));|
```


Auto Incremento

Exemplo para criação de uma tabela utilizando um campo auto incremento:

```
CREATE TABLE Tbl_Clientes(  
    Codigo INT IDENTITY(1,1) PRIMARY KEY,  
    Nome VARCHAR(100),  
    UF VARCHAR(2));
```

Nesse exemplo a tabela Tbl_Clientes possui três atributos, sendo Codigo a chave primária e tendo seu primeiro valor válido igual a 1 e se auto incrementando de 1 em 1, ou seja, primeiro registro igual a 1, o segundo igual a 2 e assim por diante.

Nota: É permitido somente um campo IDENTITY por tabela.

Auto Incremento

A inserção em tabelas que possuem campos auto incrementos deve-se suprimir da sintaxe INSERT os mesmos, conforme exemplo a seguir:

```
INSERT INTO Tbl_Clientes  
    (Nome, UF)  
VALUES  
    ('Nicholas', 'RS');|
```

Como foi demonstrado no exemplo, deve-se suprimir o atributo Codigo que é auto incremento para o valor do mesmo ser gerado automaticamente conforme suas configurações.

Auto Incremento

Para saber o valor atual do IDENTITY, basta usar o seguinte comando:

```
SELECT IDENT_CURRENT('Tbl_Clientes');
```

Ou

```
SELECT IDENT_CURRENT('Tbl_Clientes') as Ultimo_registro;
```

Será exibido o código do último registro cadastrado.

BIBLIOGRAFIA

BÁSICA:

DATE, C. J. PROJETO DE BANCO DE DADOS E TEORIA RELACIONAL: FORMAS NORMAIS E TUDO O MAIS. SÃO PAULO: NOVATEC, 2015.

ELMASRI, R.; NAVATHE, S. B. SISTEMAS DE BANCO DE DADOS: FUNDAMENTOS E APLICAÇÕES. 7 ED. SÃO PAULO: PEARSON, 2019.

HEUSER, C. A. PROJETO DE BANCO DE DADOS. 6 ED. PORTO ALEGRE: BOOKMAN, 2010.



COMPLEMENTAR:

HARRINGTON, J. L. Projeto de Bancos de Dados Relacionais: Teoria e Prática. São Paulo: Campus, 2002.

MACHADO, F. N. R., Banco de dados: projeto e implementação. 2 ed. São Paulo: Érica, 2008.

NADEAU, Tom et al. Projeto e Modelagem de Banco de Dados. 5 ed. Rio de Janeiro: Elsevier Brasil, 2013.

SILBERSCHATZ, Abraham; SUNDARSHAN, S.; KORTH, Henry F. Sistema de banco de dados. Rio de Janeiro: Elsevier Brasil, 2016.

Referências



- ALVES, W. P. FUNDAMENTOS DE BANCOS DE DADOS. ÉRICA, 2004
- HEUSER, CARLOS ALBERTO. PROJETO DE BANCO DE DADOS. SAGRA LUZZATTO, 2004.
- TEOREY, TOBY J. PROJETO E MODELAGEM DE BANCO DE DADOS. ELSEVIER, 2007.
- O.K. TAKAI; I.C.ITALIANO; J.E. FERREIRA, INTRODUÇÃO A BANCO DE DADOS
- OSVALDO KOTARO, APOSTILA, DCC-IME-USP – FEVEREIRO - 2005
- MATTOSO, MARTA, INTRODUÇÃO À BANCO DE DADOS – AULA
- GILLENSON, MARK L. FUNDAMENTOS DE SISTEMAS DE GERÊNCIA DE BANCO DE DADOS. LTC, 2006.
- BANCO DE DADOS BÁSICO, UNICAMP, CENTRO DE COMPUTAÇÃO, SLIDES.
- BOGORNÝ VANIA, MODELO ENTIDADE-RELACIONAMENTO, SLIDES.
- WWW.JOINVILLE.UDESC.BR/PORTAL/PROFESSORES/MAIA/.../6___MODELO_ER.PPT DATA DE ACESSO: 01/07/2015
- ABREU, FELIPE MACHADO; ABREU, MAURÍCIO – PROJETO DE BANCO DE DADOS – UMA VISÃO PRÁTICA - ED. ÉRICA – SÃO PAULO
- HEUSER, CARLOS ALBERTO. PROJETO DE BANCO DE DADOS – UMA VISÃO PRÁTICA. PORTO ALEGRE: SAGRA LUZZATTO, 2004.
- KORTH, H. F.; SUDARSHAN, S; SILBERSCHATZ, A. SISTEMA DE BANCO DE DADOS. 5A ED. EDITORA CAMPUS, 2006. - CAPÍTULO 6
- [HTTP://WWW.PROFTONINHO.COM/DOCS/MODELAGEM_AULA_6_ENTID_ASSOC.PDF](http://WWW.PROFTONINHO.COM/DOCS/MODELAGEM_AULA_6_ENTID_ASSOC.PDF) DATA DE ACESSO: 01/07/2015
- [HTTPS://MATERIALPUBLIC.IMD.UFRN.BR/CURSO/DISCIPLINA/4/56/1/6](https://MATERIALPUBLIC.IMD.UFRN.BR/CURSO/DISCIPLINA/4/56/1/6) DATA DE ACESSO: 01/02/2023
- ELMASRI, R.; NAVATHE S. B. SISTEMAS DE BANCO DE DADOS. 4 ED. EDITORA ADDISON-WESLEY. 2005. - CAPÍTULO 3
- DAVENPORT, THOMAS H.; PRUSAK, LAURENCE. CONHECIMENTO EMPRESARIAL: COMO AS ORGANIZAÇÕES GERENCIAM O SEU CAPITAL INTELECTUAL. RIO DE JANEIRO: CAMPUS, 1998.
- [HTTP://WWW.IME.UNICAMP.BR/~HILDETE/DADOS.PDF](http://WWW.IME.UNICAMP.BR/~HILDETE/DADOS.PDF) ACESSO EM: 12 MAIO 2016.



OBRIGADO