

BANCO DE DADOS

DML (DATA MANIPULATION LANGUAGE)

PROFº Bruno A de Moraes
bmoraes@sp.senai.br



DDL (DATA MANIPULATION LANGUAGE)

Linguagem de Manipulação de dados

- Permite alterações e exclusões de dados já existentes, bem como a adição de novos dados. Comandos:

INSERT

UPDATE

DELETE

INSERT (INSERÇÃO DE DADOS)

INSERT

Inserir uma mais informações em tabelas no banco de dados.

Sintaxe:

```
INSERT INTO <tabela> [<coluna(s)>] VALUES <valores>[:]
```



INSERT (INSERÇÃO DE DADOS)

Observações com relação ao comando INSERT:

- Colunas do tipo char e date devem ter seus valores entre aspas simples.
- O INSERT pode ser utilizado com o comando SELECT para a inclusão de valores a partir de uma outra tabela. **SENAI**

INSERT (INSERÇÃO DE DADOS)

Omitindo o nome das colunas:

- A inserção será em todas as colunas de uma linha;
- Os valores devem ser especificados na ordem em que as colunas foram definidas na tabela.

Por exemplo, inserção de um novo cliente na tabela Cliente:

Exemplo:

```
INSERT INTO CLIENTE  
VALUES(1, 'Bruno', 'Professor',  
1111-1111)
```



INSERT (INSERÇÃO DE DADOS)

Especificação explícita do nome das colunas

- Os nomes das colunas devem ser explicitamente especificados no INSERT quando:

Exemplo 1 – Existe uma ou mais colunas que permitem valores nulos e não deseja-se inserir dados nessas colunas.

Por exemplo, considerando que a coluna descricao da tabela produto permite NULL

Exemplo:

```
INSERT INTO CLIENTE (codigo, nome,  
cargo,telefone)
```

```
VALUES(1, 'Bruno',  
'Professor',1111-1111)
```



INSERT (INSERÇÃO DE DADOS)

- Especificando as colunas

```
INSERT INTO cidades (cod_cid, nome_cid) VALUES  
(1, 'Rio Claro')
```

```
INSERT INTO cidades (nome_cid, cod_cid) VALUES  
( 'Rio Claro', 1)
```

- Omitindo as colunas

```
INSERT INTO cidades VALUES (1, 'Rio Claro', 'SP')
```

```
INSERT INTO produto VALUES (1, 'Parafuso', 20)
```

```
INSERT INTO cliente VALUES (10, 'Rogerio  
Armador', 35324069)
```

ATIVIDADE 1 EXTRA (INSERT)

Crie o BD com o nome de AT1

Crie a tabela PRODUTOS (CProd, Descricao, Peso, ValorUnit)

Insira os itens abaixo:

1 , Teclado, KG, 35.00

2 , Mouse, KG, 25.00

3 , HD, KG, 350.00

ATIVIDADE 1 EXTRA (INSERT)

Crie a tabela VENDEDOR (CVend,
Nome, Salario,FSalario)

Insira os itens abaixo:

1 , Ronaldo, 3512.00, 1

2 , Robertson, 3225.00, 2

3 , Clodoaldo, 4350.00, 3

ATIVIDADE 1 EXTRA (INSERT)

Crie a tabela CLIENTE (CCli, Nome, Endereco, Cidade, UF)

Insira os itens abaixo:

11 , Bruno, Rua 1 456, Rio Claro , SP

12 , Cláudio, Rua Quadrada 234, Campinas, SP

13 , Cremilda, Rua das Flores 666, São Paulo, SP



UPDATE (ATUALIZAÇÃO DE DADOS)

UPDATE

- Atualiza os dados de uma ou mais tuplas de uma tabela.

Sintaxe:

UPDATE <tabela>

SET <coluna(s)='novo_valor'>

WHERE <condição>[:]

UPDATE (ATUALIZAÇÃO DE DADOS)

Situação: Alteração do valor de uma coluna ou mais linhas específicas da tabela.

Exemplo:

Alterar o valor unitário do produto 'teclado' de 35.00 para 32.00
:

```
UPDATE produto  
SET valorunit = 32.00  
WHERE descricao = 'teclado'
```



UPDATE (ATUALIZAÇÃO DE DADOS)

Situação: Alteração do valor de uma coluna para todos as linhas da tabela.

Exemplo:

Aumentar o salário fixo de todos os vendedores em 25%

```
UPDATE vendedor  
SET salario = (salario * 1.25)
```



UPDATE (ATUALIZAÇÃO DE DADOS)

Situação: Alteração do valor de mais de uma coluna da tabela ao mesmo tempo.

Exemplo:

Alterar o valor unitário do produto de código 2 de 25.00 para 28.00 e sua descrição de 'Mouse' para 'Mouse Branco'

```
UPDATE produto  
SET valorunit = 28.00,  
descricao = 'Mouse Branco'  
WHERE CProd = 2
```

UPDATE (ATUALIZAÇÃO DE DADOS)

- Registro único

```
UPDATE Clientes SET bairro = 'Centro' WHERE  
codCli = 'C001';
```

- Registro múltiplo

```
UPDATE produtos SET Preço = Preço + (Preço * 10%);
```

```
UPDATE      cidades      SET      cidades.uf='SP'      WHERE  
cidades.codigo=1;
```

```
UPDATE pessoas SET salario=100 adicional=20 WHERE  
categoria=1;
```

UPDATE (ATUALIZAÇÃO DE DADOS)

DESAFIO

Faça as seguintes atualizações utilizando a base de dados criada anteriormente:

1. Altere o salário dos vendedores da faixa salarial 1 para R\$ 3150,00
2. Dê 10% de aumento para os funcionários da faixa salarial 2.
3. Altere o salário dos vendedores da faixa salarial 3 para R\$ 3500,00.

Dica: Utilizem a cláusula WHERE!



DELETE (EXCLUSÃO DE DADOS)

DELETE

Remove uma ou mais tuplas de uma tabela.

Sintaxe:

```
DELETE FROM <tabela>  
WHERE <condição>[:]
```

~~DELETE (EXCLUSÃO DE DADOS)~~

Situação: Exclusão de um ou mais registros de uma tabela.

Exemplo:

Exclusão de todos os vendedores cujo salário fixo seja menos que 2500.00:

**DELETE FROM vendedor
WHERE salario < 2500.00;**

~~DELETE (EXCLUSÃO DE DADOS)~~

CUIDADO: Exclusão de todas as linhas de uma tabela!

Exemplo:

Exclusão de todas as linhas da tabela cliente:

DELETE FROM cliente;



TRUNCATE TABLE (EXCLUSÃO DE DADOS)

Outra maneira de excluir todas as linhas de uma tabela é usando o comando **TRUNCATE TABLE**. Embora seja semelhante a DELETE, ele é classificado como uma instrução DDL em vez de uma instrução DML.

Sintaxe:

```
TRUNCATE TABLE <tabela>[:]
```

Exemplo:

Exclusão de todas as linhas da tabela produto

```
TRUNCATE TABLE produto;
```

DELETE (EXCLUSÃO DE DADOS)

- Registro único

```
DELETE FROM Clientes WHERE codCli = 'C001';
```

- Registro múltiplo

```
DELETE FROM Produtos WHERE qtde = 0;
```

- Intervalo (operador AND)

```
DELETE FROM Cidades WHERE codigo >= 1 AND codigo <= 10;
```

- Esvaziando a Tabela

```
DELETE FROM Clientes;
```