

# Comandos DML

## Data Manipulation Language( Linguagem de Manipulação de Dados)

É um conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em um banco de dados.

### Insert

#### Objetivo

Incluir um novo registro em uma tabela do banco de dados.

#### Sintaxe

**Insert Into** <nome-tabela> [(<nome-coluna>,<nome-coluna>)] **values** (<relação dos valores a serem incluídos>)

Em que:

**Nome-tabela:** representa o nome da tabela na qual será incluído o registro.

**Nome-coluna:** representa o nome da(s) coluna(s) que terão conteúdo no momento da operação de inclusão.

#### Exemplos

Esse comando pode ser executado de três maneiras.

1. Quando apenas parte dos campos da tabela tiver conteúdo.

```
Insert into funcionario (idfuncionario, nome, endereço, cidade, estado, email, dataNascto) values (1, 'Jose da Silva', 'Av. São Paulo, 34', 'Itu', 'SP', 'zesilva@yahoo.com.br', '1991-02-24')
```

Neste caso devem ser especificadas todas as colunas que terão conteúdo e os valores relacionados devem obedecer a essa sequencia. Para os campos que não tem conteúdo especificado será preenchido o valor null.

2. Quando todos os campos da tabela tiverem conteúdo.

```
Insert into funcionario values (2, 'Joao de Oliveira', 'Rua da Saude, 138', 'Jundiai', 'SP', 'jojo@hotmail.com', '1988-01-11', 750, 'TI', 1)
```

Neste caso não é necessário especificar as colunas, entretanto a relação dos valores a serem incluídos deve obedecer a mesma sequência da definição da tabela.

3. Varias linhas ao mesmo tempo.

```
Insert into funcionario values ( 3, 'Ana Pimenta', 'Rua da Saude,11' , 'Jundiai', 'SP',  
'pimentinha@hotmail.com', '1987-01-01', 750, 'TI',0 ), (4, 'Mario Lopes', 'Rua Angelica,  
143', 'Jundiai', 'SP', 'mlopes@hotmail.com', '1985-04-10', 750, 'TI', 0)
```

Neste caso dois funcionários são incluídos juntos, ou seja, no mesmo insert.

## Update

### Objetivo

Atualiza os dados de um grupo de registros em 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>
```

Em que:

**nome-tabela:** representa o nome da tabela cujo conteúdo será alterado.

**Nome-coluna:** representa o nome da(s) coluna(s) que terão seus conteúdos alterados com o novo valor especificado.

**Condição:** representa a condição para a seleção dos registros que serão atualizados. Essa seleção pode resultar em um ou vários registros. Neste caso a alteração ocorre em todos os registros selecionados.

### Exemplos

```
Update funcionario set cidade = 'Jundiai'
```

Altera para Jundiai todas as cidades.

```
Update funcionario set cidade = 'valinhos' where cidade = 'Itu'
```

Altera para Valinhos somente a cidade Itu.

## Delete

### Objetivo

Deletar um registro ou um grupo de registros em uma tabela do banco de dados.

### Sintaxe

**Delete from** <nome-tabela> **where** <condição>

Em que:

**Nome-tabela:** representa o nome da tabela cujos registros serão deletados.

**Condição:** representa a condição para a deleção dos registros. A seleção pode resultar em um ou vários registros. Neste caso a operação ocorre em todos os registros selecionados.

### Exemplos

**Delete from** funcionários

Exclui todos os funcionários.

**Delete from** funcionário **where** idfuncionario = 2

Exclui somente o funcionário 2.

## Select

### Objetivo

Selecionar um conjunto de registros em uma ou mais tabelas que atenda a uma determinada condição definida pelo comando.

### Sintaxe

**Select** [**distinct** ] | **all**< nome-coluna> [, <nome-coluna>]

**From** <nome-tabela> [, <nome-tabela>]

**Where** <condição>

**Group by** <nome-coluna>

**Having** <condição>

**Order by** <nome-campo> asc | desc

Em que:

**Nome-tabela:** representa o nome da(s) tabela(s) que contem as colunas que serão selecionadas ou utilizadas para a execução da consulta.

**Condição:** representa a condição para a seleção dos registros. Essa seleção pode resultar em um ou vários registros.

**Nome-coluna:** representa a(s) coluna(s) cujos resultados são agrupados para atender a consulta.

**All:** argumento default. Mostra todos os valores obtidos na seleção.

**Distinct:** argumento que mostra os valores obtidos na seleção, eliminando as duplicidades.

**Where:** especifica o critério de seleção dos registros nas tabelas especificadas.

**Group by:** especifica o(s) campo(s) que serão agrupados para atender a consulta.

**Having:** especifica uma condição para a seleção de um grupo de dados. Esse argumento só é utilizado combinado com o argumento group by.

**Order by:** esse argumento, quando utilizado, apresenta o resultado da consulta ordenado de forma crescente (asc – opção default) ou decrescente (desc) pelos campos definidos.

### Exemplos

**Select \* from** funcionário

Exibe todos os dados dos funcionários.

**Select \* from** funcionário **order by** nome

Exibe todos os dados dos funcionários ordenados por nome.

**Select** nome **from** funcionário **order by** nome desc.

Exibe somente os nomes dos funcionários em ordem decrescente.

**Select** nome, email **from** funcionário **where** cidade = 'Valinhos'.

Exibe somente o nome e o email dos funcionários da cidade de Valinhos.

**Select** estado **from** funcionário

Exibe somente o estado dos funcionários, sem repetir estados iguais.

Utilize a tabela funcionário (BD Recursos Humanos), elaborada no decorrer do capítulo 1.

1. Insira os dados apresentados em seguida.

IdFuncionario	nome	endereço	cidade	estado	Email	dataNasc
5	Carlos Dias	Av. Lapa,121	Itu	SP	<a href="mailto:carlao@gmail.com">carlao@gmail.com</a>	31/03/1990
6	Ana Maria da Cunha	Av.São Paulo,388	Itu	SP	<a href="mailto:aninhacunha@gmail.com">aninhacunha@gmail.com</a>	12/04/1988
7	Claudia Regina Martins	Rua Holanda,89	Campinas	SP	<a href="mailto:cregina@gmail.com">cregina@gmail.com</a>	04/12/1988
8	Marcela Tatho	Rua Bélgica,43	Campinas	SP	<a href="mailto:marctatho@gmail.com">marctatho@gmail.com</a>	09/11/1987
9	Jorge Luis Rodrigues	Av. da Saudade,1989	São paulo	SP	<a href="mailto:jorgeluis@yahoo.com.br">jorgeluis@yahoo.com.br</a>	05.05.1990
10	Ana Paula Camargo	Rua Costa e Silva,33	Jundiai	SP	<a href="mailto:apcamargo@gmail.com">apcamargo@gmail.com</a>	30.06.1991
11	Ivo Cunha	Av. Raio de Luz,100	Campinas	SP	<a href="mailto:ivo@bol.cim.br">ivo@bol.cim.br</a>	11/04/1987
12	Carlos Luis de Souza	Rua Nicolau Coelho,22	São paulo	SP	<a href="mailto:cls@bol.com.br">cls@bol.com.br</a>	30/04/1988

2. Altere para Valinhos os funcionários da cidade de Itu
3. Altere os valores das colunas cargo e salário de acordo com cidade.

Cargo	Salario	Cidade
AI	1100	Valinhos
PC	1700	Campinas
TI	750	Jundiai

4. Exiba nome e cargo de todos os funcioanrios.
5. Exiba idFuncionario e email dos funcionários do estado de SP
6. Apague o funcionário 5
7. Exiba a cidade e o estado de todos os funcionários com cargo “PC”, sem repeti-los.