

Comandos DML

SQL – COMANDOS BÁSICOS

- Comando da Linguagem de Manipulação de Dados(DML)
- Permite a visualização e alteração do conteúdo dos dados de tabelas básicas.
- INSERT – Inclusão de uma ou várias tuplas em uma tabelas
- SELECT – Seleção de atributos de uma tabela
- UPDATE – Atualização de valores de atributos
- DELETE – Remoção de uma ou várias tuplas de uma tabela

Comando INSERT

- INSERT INTO nome_tabela (n_coluna1, n_coluna2, ...) VALUES ('valor1', 'value', ...);
- INSERT INTO – a palavra chave *inicia a declaração*
- Logo depois o nome_tabela a qual os dados serão inseridos
- Entre parênteses o nome das colunas que compoem esta tabela
- VALUES indica que valores serão inseridos
- ('value1' , 'value2' , ...) os valores serão inseridos de acordo com as colunas.
- **Obs: os valores precisam estar na mesma ordem que os nome das colunas**
- As aspas simples são utilizadas para inserir caracteres, textos
- Os números são escritos sem aspas

COMANDO SELECT

- SELECT – especifica as colunas da tabela
- FROM – especifica as tabelas;
- WHERE – especifica as linhas;
- *O Comando SELECT elimina os resultados para nós e exibe somente as linhas que são compatíveis com a condição estabelecida.*
- *A cláusula WHERE permite que o sistema busca algo específico*

COMANDO SELECT - Sintaxe

A. *Selecionando Colunas Específicas da Tabela*

- SELECT <nome(s) da(s) coluna(s)
 - FROM <tabela>;
-
- Exe.: Selecionar o valor e a descrição de cada produto.
 - **SELECT** valor_produto, descricao_produto **FROM** produto;

COMANDO SELECT - Sintaxe

B. Selecionando todas as colunas da tabela

- `SELECT *FROM <nome_da_tabela>;`
- O **(asterisco)* indica que todas as colunas serão selecionadas.
- *Ex.: Selecionar todas as colunas da tabela cliente*
- **`SELECT *FROM cliente;`**

Comando SELECT + CLAUSULA WHERE

C. SELECIONANDO APENAS ALGUMAS LINHAS DA TABELA

- A clausula WHERE em um comando SELECT especifica quais linhas queremos obter, baseada em condições de seleção.
- Sintaxe básica.
 - SELECT <nome(s) da(s) coluna(s)>
 - FROM <nome da tabela>
 - WHERE <condições de seleção>;

Comparações na Cláusula – WHERE

- **SELECIONANDO APENAS ALGUMAS LINHAS DA TABELA**
- WHERE <nome da coluna> <operador> <valor>;
 - Operadores de Comparação

OPERADOR	VALOR
=	Igual
<> Ou !=	Diferente
<	Menor que
>	Maior que
>=	Maior ou igual do que
<=	Menor ou igual do que
!>	Não maior
<!	Não menor

Comparações na Cláusula - WHERE

- WHERE <nome da coluna> <operador> <valor>;
- Exe1.:
 - Listar o número do pedido, o código do produto e a quantidade dos itens de pedido com a quantidade igual a 35 da tabela item do pedido.
 - SELECT num_pedido, codigo_produto, quantidade
 - FROM item_de_pedido
 - WHERE quantidade = 35;

Comparações na Cláusula - WHERE

Exe2.: Litar todos os clientes que moram em Niterói

```
SELECT nome_cliente  
FROM cliente  
WHERE cidade = 'niteroi';
```

Obs: as aspas simples sempre tem que ser colocadas para trazer os resultados de caracteres; esquecer de colocas pode não trazer os resultados ou não trazer nada.

CLÁUSULA - WHERE

D. Operadores Lógicos

OPERADOR	VALOR
AND	E – Lógico
OR	OU – lógico
NOT	negação

Exe.: Listar os produtos que tenham unidade igual a 'M' e valor unitário igual a R\$ 1,05 da tabela produto.

```
SELECT descricao_produto  
FROM produto  
WHERE unidade = 'M' AND valor_do_produto = 1.05;
```

Clausula - WHERE

- ***Operadores Lógicos***
- Exe3.: listar os clientes e seus respectivos endereços, que moram em 'SÃO PAULO' ou estejam na faixa de CEP entre 30077000 e 30079000. classificado por nome de ordem.
- **SELECT** nome_cliente, endereco
- **FROM** cliente
- **WHERE** (cep >= 30077000 AND cep <= 30079000) OR cidade = 'São Paulo';

Clausula - WHERE

- ***Operadores Lógicos***

- Exe4.: listar os pedidos que não tenham prazo de entrega igual a 15 dias

- **SELECT** codigo_do_pedido

- **FROM** pedido

- **WHERE NOT** (prazo_entrega = 15);

- Ou poderia utilizar o operador de comparação(diferente <> ou !=)

- **SELECT** num_pedido

- **FROM** pedido

- **WHERE** (prazo_entrega != 15);

CLÁUSULA - WHERE

D. Ordenando os dados selecionados

Sintaxe básica:

SELECT <nome da(s) coluna(s)>

FROM <tabela>

WHERE <condição(ões)>

ORDER BY <nome da(s) colunas(s)> | **ASC**
DESC

- Ou
- **ORDER BY** <número da coluna>
- **Obs.:** quando a palavras **ASC** OU **DESC** forem omitidas o valor da consulta por padrão será **ASC**.

CLÁUSULA - WHERE

- ***Ordenando os dados selecionados***
- Exe5.: listar em ordem alfabética todos os seus vendedores e seus respectivos salários
- **SELECT** nome_vendedor, salario_fixo
- **FROM** vendedor
- **ORDER BY** nome_vendedor;

CLÁUSULA - WHERE

- **Ordenando os dados selecionados**
- Exe6.: listar os nomes, cidades e estados de todos os clientes, ordenados por estado e estado de forma descendente
- **SELECT** nome, cidade, uf
- **FROM** cliente
- **ORDER BY** UF **DESC**, cidade **DESC**;
- Exe7.: Mostrar a descrição e o valor unitário de todos os produtos que tenham a unidade 'KG', em ordem de valor unitário ascendente.
- **SELECT** descricao, valor_produto
- **FROM** produto
- **WHERE** unidade = 'KG'
- **ORDER BY** valor_produto **ASC**;

CLÁUSULA - WHERE

E. Manipulando dados numéricos

Operadores aritméticos podem ser usados sobre qualquer coluna numérica, incluindo colunas de tipo de dado, int, dec, float e real.

Os operadores aritméticos são:

SÍMBOLO	OPERAÇÃO	TIPO DE DADO
+	Adição	Int, dec, float e real
-	Subtração	Int, dec, float e real
/	Divisão	Int, dec, float e real
*	Multiplicação	Int, dec, float e real
%	Módulo	int

Clausula - WHERE

- ***Manipulando dados numéricos***
- ***Operado adição (+)***
- *Exe8.: selecionar todos os vendedores com salário fixo e somar R\$ 100,00 de gratificação para cada um deles.*
- ***SELECT*** nome_vendedor, salario_fixo = (salario_fixo + 100)
- ***FROM*** vendedor;

Clausula - WHERE

- **Manipulando dados numéricos**
- **Operado subtração (-)**
- *Exe9.: selecionar todos os vendedores com salário fixo e descontar R\$ 100,00 por faltas no trabalho.*
- **SELECT** nome_vendedor, salario_fixo = (salario_fixo - 100)
- **FROM** vendedor;

Clausula – WHERE com LIKE

- Comando **LIKE** é utilizado quando precisamos buscar um determinado texto dentro de um campo com valores textuais. Seria uma espécie de “filtro”.
- Sintaxe: **SELECT** colunas **FROM** tabela **WHERE** campo **LIKE** ‘valor’
- Nessa Instrução, ‘valor’ pode ser informado de várias formas:
 - **valor**: serão retornados todos os registros que contenham **EXATAMENTE** o texto ‘valor’;
 - **%valor%**: serão retornados os registros que contenham o ‘valor’ informado, podendo aparecer em qualquer parte do campo buscado;
 - **%valor**: serão retornados os registros que terminem com o ‘valor’ pesquisado, independente de qual texto esteja começando;
 - **valor%**: serão retornados os registros que comecem com o ‘valor’ pesquisado, independente de qual texto esteja terminando;

Clausula – WHERE com BETWEEN

- Comando **BETWEEN** é utilizado quando precisamos recuperar registros na tabela cujo valor de um campo encontra-se em um intervalo especificado. Muito usado para filtrar dados por intervalo de datas.
- Sintaxe: **SELECT** colunas **FROM** tabela **WHERE** campo **BETWEEN** inicio_intervalo **AND** fim_intervalo
- Exemplos:
 - **SELECT** * **FROM** Pessoa **WHERE** Nascimento **BETWEEN** '01-01-1981' **AND** '31-12-1990'
 - **SELECT** * **FROM** Produto **WHERE** preco **BETWEEN** 20 **AND** 40

Comando - UPDATE

- **Atualizando um Registro – UPDATE**
- A atualização de dados em linhas existentes na tabela permite que:
- Especifique-se uma determinada coluna e altere-se seu valor.
- Seja indicada uma linha específica ou uma condição de identificação de linhas para que sejam alterados valores de determinadas colunas.
- Sintaxe:
- **UPDATE** <nome da tabela>
 - **SET** <nome da(s) coluna(s)> = valor
 - **WHERE** <condição>;

Comando - UPDATE

- **Atualizando um Registro – UPDATE**

Ex1.: Alterar o valor unitário do produto 'parafuso' de R\$ 1.25 para R\$ 1.62.

- **UPDATE** produto
 - **SET** val_unit = 1.62
 - **WHERE** descricao = 'Parafuso';
- **SET** – informa o novo valor ao sistema
- **A clausula WHERE** informa a linha que será alterada.

Comando - DELETE

- **Apagando Registros da Tabela**
- Sintaxe:
- **DELETE FROM** <nome da tabela>
 - **WHERE** <condicao>;
 - Exe.: apagar todos os clientes que moram em 'São Paulo'.
 - **DELETE FROM** cliente
 - **WHERE** cidade = 'São Paulo';
- **Obs.: se a cláusula WHERE for omitida todos os registros da tabela serão apagados**