

# Banco de Dados Relacional

PROF<sup>o</sup> DR<sup>o</sup> FRANCISCO DOUGLAS. L ABREU

MSC E PHD EM ENGENHARIA BIOMÉDICA  
ESP. EM ANÁLISE DE DADOS

FRANCISCO.ABREU@FATEC.SP.GOV.BR





# LINGUAGEM DE MANIPULAÇÃO DE DADOS (DML)





**DDL**

**CREATE  
ALTER  
DROP  
TRUNCATE**

**DML**

**INSERT  
DELETE  
UPDATE**

**DQL**

**DTL**

**DCL**




# Linguagem de Manipulação de Dados (DML)

Por manipulação de dados entendemos:

- **Inserção de dados;**
- **Remoção de dados;**
- **Modificação de dados.**

A DML é a linguagem que viabiliza o acesso ou a manipulação dos dados de forma compatível ao modelo de dados apropriado.

São basicamente dois tipos:

- **DML Procedural:** exige que o usuário especifique quais dados são necessários, e como obtê-los;
  - **DML Não-Procedural:** exige que o usuário especifique quais dados são necessários, sem especificar como obtê-los. Mais fácil de aprender a usar. Pode gerar códigos não muito eficientes
- 



# Apagar Dados (DELETE)

O comando *DELETE* **apaga uma ou mais tuplas (linhas)** específicas em uma tabela do banco de dados, principalmente com a cláusula *WHERE* especificada.

Se a cláusula *WHERE* estiver ausente, o efeito de exclusão ocorrerá em todas as tuplas da tabela e **os efeitos poderão ser irreversíveis**. Para evitar a exclusão permanente, utiliza-se o *Transact* em SQL.

## SINTAXE

```
DELETE FROM <NOME_DO_BANCO>.<NOME_DA_TABELA>  
WHERE [CONDIÇÃO];
```





# Apagar Dados (DELETE)

## Parâmetros:

### <NOME\_DA\_TABELA>

O nome da tabela onde os dados serão apagados;

### <WHERE CONDIÇÃO>

condição de consulta do SQL que retorna as linhas a serem excluídas





## Exemplos:

**delete from** cidades **where** cod\_cid=1;

**delete from** cidades **where** nome\_cid='São Paulo';

**delete from** cidades **where** data\_criacao='2021-10-09';





# Alterar Dados (UPDATE)

O comando *UPDATE* **altera uma ou mais colunas em tuplas (linhas)** específicas em uma tabela do banco de dados, principalmente com a cláusula *WHERE* especificada.

Se a cláusula *WHERE* estiver ausente, o efeito de atualização ocorrerá em todas as tuplas da tabela e **os efeitos poderão ser irreversíveis**. Para evitar a atualização permanente, utiliza-se o *Transact* em SQL.

Somente as colunas a serem modificadas devem aparecer na lista de colunas atribuição (*SET*).

## SINTAXE

**UPDATE**<NOME\_DO\_BANCO>.<NOME\_DA\_TABELA>

**SET** COLUNA=NOVO\_VALOR

**WHERE** [CONDIÇÃO];







# Alterar Dados (UPDATE)

## Parâmetros:

### <NOME\_DA\_TABELA>

O nome da tabela onde os dados serão apagados;

### <COLUNA>

O nome da coluna da tabela;

### <EXPRESSÃO>

Uma expressão válida ou um valor a ser atribuído a coluna;

### <WHERE CONDIÇÃO>

condição de consulta do SQL que retorna as linhas a serem excluídas






## Exemplos:

**UPDATE** cidades **set** nome\_cid='São Paulo' **where** cod\_cid=1;

**UPDATE** cidades **set** nome\_cid='SÃO PAULO' **where** nome\_cid='San Paul';

**UPDATE** cidades **set** data\_criacao='2021-10-10' **where** data\_criacao='2021-10-09';





# ATIVIDADE EM AULA

## CRIE UMA TABELA:

```
CREATE TABLE ALUNO (  
  ID INT,  
  NOME VARCHAR (50),  
  MEDIA DOUBLE,  
  ANO DATE  
);
```

## INSERE DADOS NA TABELA:

```
INSERT INTO ALUNO VALUES(1, "João", 5, "2020" );  
INSERT INTO ALUNO VALUES(2, "Maria",10, "2020" );  
INSERT INTO ALUNO VALUES(3, "Aline", 2, "2011" );  
INSERT INTO ALUNO VALUES(4, "Pedro", 5, "2012" );
```






# WHERE

A clausula **WHERE** cria uma condição para que a operação seja efetuada caso o registro atenda a condição imposta.

Ou seja, ela pode ser utilizada para Filtrar os dados de uma consulta (**SELECT**), para impor uma restrição em uma exclusão (**DELETE**) e para impor restrições nos registros atualizados (**UPDATE**).

As condições criadas podem ser combinadas com alguns comandos ou com expressões simples:



# OPERADORES

SIMBOLO	SIGNIFICADO
<b>=</b>	<b>IGUAL A (IGUALDADE)</b>
<b>&lt;&gt;</b>	<b>DIFERENTE A (DESIGUALDADE)</b>
<b>&lt;</b>	<b>MENOR QUE</b>
<b>&gt;</b>	<b>MAIOR QUE</b>
<b>&gt;=</b>	<b>MAIOR OU IGUAL A QUE</b>
<b>&lt;=</b>	<b>MENOR OU IGUAL A QUE</b>



## **AND** (Lógica Booleana E)

Ex: **UPDATE** ALUNO **SET** ano=2020, nome='João' **WHERE** nome='Pedro' **AND** ano=2015 ;

**Resultado Esperado:** Nenhuma linha (tupla) será Alterada.

**Explicativo:** Será alterado apenas a coluna que atenda a todas condições impostas (Todos os valores tem que ser verdadeiros/satisfeitas a condição) :

Ex: **DELETE FROM** Aluno **WHERE** ano=20202 **AND** nome='Pedro';

**Resultado Esperado:** Duas linhas (tuplas) serão removidas .

**Explicativo:** : A verificação das condições só será verdadeira quando qualquer uma das condições forem satisfeitas.





## **OR** (Lógica Booleana OU)

Ex: **UPDATE** Aluno **SET** **WHERE** ano=2020 **OR** nome='João';

**Resultado Esperado:** Duas linhas (tupla) serão alteradas.

**Explicativo:** : A verificação das condições só será verdadeira quando qualquer uma das condições forem satisfeitas.



# Atividade em Aula

EXECUTE O SCRIPT ABAIXO:

```
CREATE TABLE cliente (  
    cod_cli int,  
    nome varchar(50),  
    cpf varchar(14),  
    rua varchar(50),  
    num varchar(5),  
    bairro varchar(50),  
    cidade varchar(50),  
    uf varchar(2),  
    cep varchar(9),  
    PRIMARY KEY (cod_cli)  
);
```



# Atividade em Aula

EXECUTE OS COMANDOS ABAIXO ANOTANDO OS RESULTADOS. EM CASO DE ERRO, EXPLIQUE O MOTIVO E FAÇA A CORREÇÃO NECESSÁRIA:



```
delete from cliente where cod_cli<0;  
  
delete from cliente where nome='bruno';  
  
delete from cliente;
```

# Atividade em Aula

EXECUTE OS COMANDOS ABAIXO ANOTANDO OS RESULTADOS. EM CASO DE ERRO, EXPLIQUE O MOTIVO E FAÇA A CORREÇÃO NECESSÁRIA:



```
UPDATE CLIENTE SET NOME='JOÃO' WHERE COD_CLI=0;
```

```
UPDATE CLIENTE SET NOME='JOÃO' WHERE NOME='BRUNO';
```

**Alguma Dúvida?**