



SQL

DML: *Data Manipulation Language*

Juliano Varella de Carvalho
Professor



Universidade Feevale | +55 51 3586.8800 • www.feevale.br



SQL: DML

- **DML** permite manipular os dados
 - Buscar os dados: *select*
 - Inserir os dados: *insert*
 - Atualizar os dados: *update*
 - Excluir os dados: *delete*

SQL: DML

- Cláusula genérica

```
SELECT  $A_1, A_2, A_3, \dots, A_n$   
FROM  $r_1, r_2, \dots, r_m$   
WHERE  $P$ 
```

- a cláusula ***select*** captura os atributos desejados no resultado da consulta;
- a cláusula ***from*** corresponde as relações sobre as quais serão realizadas as pesquisas;
- a cláusula ***where*** consiste em um predicado envolvendo atributo(s) da relação(ões) que aparece(m) na cláusula *from*.

SQL: DML

Os exemplos das cláusulas em DML serão aplicados sobre as três tabelas abaixo: *peça*, *fornecedor* e *expedição*.

PEÇA				
CODP	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Hammer	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

EXPEDICAO		
CODS	CODP	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

SQL: DML

PECA				
CODP	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Hammer	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

EXPEDICAO		
CODS	CODP	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Tabela PECA: mantém dados sobre os tipos de peças. Tem como atributos:

- CODP: número da peça (único);
- PNAME: nome da peça;
- COLOR: cor;
- WEIGHT: peso em libras;
- CITY: localidade onde são armazenadas as peças.

Tabela FORNECEDOR: mantém dados sobre os fornecedores. Tem como atributos:

- CODS: número único de fornecedor;
- SNAME: nome do fornecedor;
- STATUS: um valor de condição;
- CITY: localização do fornecedor

Tabela EXPEDICAO: representa as expedições de uma peça por um fornecedor. Tem como atributos:

- CODP: número da peça;
- CODS: número do fornecedor;
- QTY: quantidade de peças expedidas

Select: iniciando

Obter números e status dos fornecedores de Paris

```
SELECT CODS, STATUS  
FROM FORNECEDOR  
WHERE CITY = 'Paris';
```

Consulta com nomes de campos qualificados

```
SELECT FORNECEDOR.CODS, FORNECEDOR.STATUS  
FROM FORNECEDOR  
WHERE FORNECEDOR.CITY = 'Paris';
```

Resposta de ambas consultas


CODS	STATUS
S2	10
S3	30

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Select: iniciando

Obter os números de todas as peças expedidas

```
SELECT CODP  
FROM EXPEDICAO;
```



CODP
P1
P2
P3
P4
P5
P6
P1
P2
P2
P2
P4
P5

EXPEDICAO		
CODS	CODP	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Eliminando a duplicação das linhas

```
SELECT DISTINCT CODP  
FROM EXPEDICAO;
```




CODP
P2
P6
P4
P3
P5
P1

Select: iniciando

PECA				
CODP	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Hammer	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Obter o número da peça e o peso da mesma em gramas, de todas as peças"

```
SELECT CODP, WEIGHT * 454  
FROM PECA;
```




CODP	WEIGHT * 454
P1	5448
P2	7718
P3	7718
P4	6356
P5	5448
P6	8626

Rotulando a expressão

```
SELECT CODP, WEIGHT * 454 as "Peso em gramas"  
FROM PECA;
```

Ou

```
SELECT CODP, WEIGHT * 454 "Peso em gramas"  
FROM PECA;
```



CODP	Peso em gramas
P1	5448
P2	7718
P3	7718
P4	6356
P5	5448
P6	8626

Select: iniciando

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Recuperando todos os atributos de uma só vez.

Para isto pode-se utilizar o operador asterisco (*). Obter todos os registros de todos os atributos.

```
SELECT *  
FROM FORNECEDOR;
```

Ou

```
SELECT CODS, SNAME, STATUS, CITY  
FROM FORNECEDOR
```



CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Select: iniciando

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Recuperando valores com base em um **predicado**: "Obter os números dos fornecedores de Paris com Status > 20"

```
SELECT  CODS
FROM    FORNECEDOR
WHERE   CITY = 'Paris'
        AND
        STATUS > 20;
```



CODS
S3

O **predicado** pode incluir operadores comparativos: =, <>, >=, <, <= e operadores lógicos: AND, OR e NOT.

Select: iniciando

PECA				
CODP	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Hammer	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Observando a importância de utilizar parênteses para indicar a ordem de avaliação da cláusula.

```
SELECT *  
FROM PECA  
WHERE
```

CITY = 'Paris' AND (WEIGHT >= 15 OR COLOR = 'Blue')



CODP	PNAME	COLOR	WEIGHT	CITY
P2	Bolt	Green	17	Paris
P5	Cam	Blue	12	Paris

```
SELECT *  
FROM PECA  
WHERE
```

CITY = 'Paris' AND WEIGHT >= 15 OR COLOR = 'Blue'



CODP	PNAME	COLOR	WEIGHT	CITY
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P5	Cam	Blue	12	Paris


Select: iniciando

FORNECEDOR			
CODS	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Trabalhando com ordenação dos registros, baseado em alguns atributos.
Obter os números e status dos fornecedores de Paris, na ordem decrescente de status

```
SELECT CODS, STATUS  
FROM FORNECEDOR  
WHERE CITY = 'Paris'  
ORDER BY STATUS DESC;
```

```
SELECT CODS, STATUS  
FROM FORNECEDOR  
WHERE CITY = 'Paris'  
ORDER BY 2 DESC;
```



CODS	STATUS
S3	30
S2	10

DESC: ordena de modo decrescente
ASC: ordena de modo crescente (*default*)

A ordenação é especificada pela cláusula
ORDER BY

Insert

Operador que tem como função inserir registros dentro de uma tabela.

INSERT

INTO tabela [(campo1 [, campo2] ...)]
VALUES (constante1 [, constante2] ...);

INSERT

INTO tabela [(campo1 [, campo2] ...)]
SELECT ... FROM ... WHERE;

No **primeiro caso**, insere-se uma linha na “tabela”, que tem os valores especificados para cada campo listado, ou seja, a *constante i* da lista de constantes corresponde ao *campo i* da lista de campos.

No **segundo caso**, o conjunto resultado da operação SELECT é inserido dentro de “tabela”

Insert: exemplos

- **Inserindo registro único com INSERT**

Acrescentar a peça P7, com os seguintes valores: cidade igual a Athens, peso igual a 24 libras, nome igual a Pliers e cor 'Blue'.

```
INSERT INTO PECA (CODP, CITY, WEIGHT, PNAME, COLOR)
VALUES ('P7', 'Athens', 24, 'Pliers', 'Blue');
```

- **Registro único INSERT com omissão dos nomes de campo:**

Acrescentar a peça P8 (nome: Drill, cor Green, peso 14, cidade Nice).

```
INSERT INTO PECA
VALUES ('P8', 'Drill', 'Green', 14, 'Nice');
```

Insert: exemplos

- Inserindo registro único com INSERT

Inserir uma nova expedição com número de fornecedor S20, número da peça P20 e quantidade 1000.

```
INSERT  
INTO EXPEDICAO (CODS, CODP, QTY)  
VALUES ('S20', 'P20', 1000);
```

Erro de Integridade Referencial

ERRO: inserção ou atualização em tabela "expedicao" viola restrição de chave estrangeira "expedicao_cods_fkey"
DETAIL: Chave (cods)=(S20) não está presente na tabela "fornecedor".

ERRO: inserção ou atualização em tabela "expedicao" viola restrição de chave estrangeira "expedicao_codp_fkey"
DETAIL: Chave (codp)=(P20) não está presente na tabela "peca".


Insert: exemplos

- Inserindo múltiplos registros com INSERT

Supondo que exista uma tabela denominada *TEMP* (codp:texto, totqty:inteiro)
De cada peça fornecida (expedida), obter o número da peça e a quantidade total fornecida e, de uma vez só, armazenar os valores na tabela TEMP.

```
INSERT  
INTO TEMP (CODP, TOTQTY)  
  SELECT CODP, SUM(QTY)  
  FROM EXPEDICAO  
  GROUP BY CODP;
```

SELECT * FROM TEMP;



TEMP	
CODP	TOTQTY
P2	1000
P6	100
P4	500
P3	400
P5	500
P1	600

Delete

Operador que tem como função remover registros de dentro de uma tabela.

```
DELETE  
FROM tabela  
[WHERE predicado];
```

Eliminam-se todos os registros da “tabela” que satisfaçam o “predicado”.

Se não houver um predicado, todos os registros da tabela são removidos.

Não há como fazer operações DELETE em diversas tabelas.

Delete: exemplos

- Removendo um registro único com DELETE

Eliminar o fornecedor 'S1'.

```
DELETE  
FROM FORNECEDOR  
WHERE CODS = 'S1';
```

- Removendo registros múltiplos com DELETE

Apagar todos os fornecedores da cidade de 'Madrid'.

```
DELETE  
FROM FORNECEDOR  
WHERE CITY = 'Madrid';
```

Delete: exemplos

- Removendo registros múltiplos com DELETE

Eliminar todas as expedições realizadas.

```
DELETE  
FROM EXPEDICAO;
```

- Usando DELETE com subconsulta

Apagar todas as expedições de fornecedores de Londres

```
DELETE FROM EXPEDICAO  
WHERE 'London' = (  
    SELECT CITY  
    FROM FORNECEDOR  
    WHERE FORNECEDOR.CODS = EXPEDICAO.CODS);
```

Update

Operador que tem como função atualizar os registros que já estão dentro de uma tabela.

```
UPDATE tabela  
    SET campo1 = expressão1  
    [, campo2 = expressão2] ...  
    [WHERE predicado];
```

Modificam-se todos os registros de “tabela” que satisfaçam o “predicado”, de acordo com as indicações (“campo = expressão”) na cláusula SET.

Se não houver um predicado, todos os registros da tabela são atualizados.

Não há como fazer operações UPDATE em diversas tabelas.

Update: exemplos

- **Atualizando um registro único com UPDATE**

Modificar a cor da peça 'P2' para 'Red', aumentar o peso em 5 e determinar a cidade como "desconhecida"

```
UPDATE PECA  
SET COLOR = 'Red', WEIGHT = WEIGHT + 5, CITY = NULL  
WHERE CODP = 'P2';
```

- **Atualizando múltiplos registros com UPDATE**

Dobrar o status de todos os fornecedores de Londres

```
UPDATE FORNECEDOR  
SET STATUS = STATUS * 2  
WHERE CITY = 'London';
```

Update: exemplos

- Atualizando registros utilizando UPDATE

Ajustar a quantidade de expedição para zero em todos os fornecedores de Londres.

```
UPDATE EXPEDICAO
SET QTY = 0
WHERE 'London' = (
    SELECT CITY
    FROM FORNECEDOR
    WHERE FORNECEDOR.CODS = EXPEDICAO.CODS);
```