

COMANDO SELECT

Este comando é usado para fazer consultas a um conjunto de Tabelas, sem alterar o conteúdo dessas Tabelas originais.

Ele gera uma Tabela resposta à consulta.

VAMOS DIVIDIR NOSSO ESTUDO EM 2 PARTES

- A) **SELECT** usado para consulta a uma só tabela.
- B) **SELECT** usado para consulta a mais de uma
tabela

Observações gerais sobre o SELECT

- É um só comando, mas composto de cláusulas.
- Cada cláusula deve ser escrita em linhas diferentes (não é obrigatório, mas aumenta a clareza do comando).
- Exige um “;” após a última cláusula.
- Cada cláusula executa uma tarefa específica dentro do comando.
- Os símbolos < e > não devem constar do comando. Eles devem ser substituídos pelo que está informado entre eles.

USAREMOS AS TABELAS ABAIXO PARA EXEMPLIFICAR O USO DO COMANDO

TABELAS:

ENTIDADE 1



TAB ASP			
NIP	NOMEA	SEXO	ANO
1	PAULO	M	1
2	PAULA	F	1
3	PEDRO	M	2
4	HELENA	F	1
5	SERGO	M	1

ENTIDADE 2



TAB PRODUTO		
CODP	NOMEP	PUNIT
1	PAO	2.00
2	LEITE	5.00
3	CAFÉ	7.00
4	SUCO	9.00
5	QUEJO	5.00

RELACIONAMENTO



TAB COMPRA			
NIP	CODP	DATA	VCOMP
1	2	05/07/2017	15.00
1	3	07/10/2017	7.00
1	5	10/10/2017	10.00
2	1	15/09/2016	8.00
2	3	16/10/2017	14.00
4	1	20/09/2017	4.00
4	5	23/09/2017	10.00
5	1	27/09/2017	2.00

FORMATOS

NIP	INT
SEXO	STRING
ANO	INT
CODP	INT
PUNIT	REAL
VCOMP	REAL
NOMEA	STRING
NOMEP	STRING
DATA	STRING

A) SELECT - consulta a uma só tabela

- 1) Forma mais simples (cláusulas obrigatórias).

SELECT <nome dos atributos a serem mostrados>

FROM < nome da tabela de consulta >;

Funcionamento desse comando:

Monta uma tabela só com os atributos (colunas) selecionados da tabela escolhida.

Mostra todas as linhas da tabela escolhida para atributos selecionados.

Exemplos:

1.1 select NOMEA, SEXO
from TABASP;

Resultado:

NOMEA	SEXO
PAULO	M
PAULA	F
PEDRO	M
HELENA	F
SERGIO	M

1.2 select *
from TABASP;

Resultado: Mostra toda a tabela TABASP.

O símbolo * representa todos os atributos da tabela

A) SELECT - consulta a uma só tabela

2) Cláusula WHERE.

SELECT <nome dos atributos a serem mostrados>

FROM < nome da tabela de consulta >

WHERE <condições>;

Funcionamento desse comando:

Monta uma tabela só com os atributos (colunas) selecionados da tabela escolhida.

Só mostra as linhas da tabela escolhida que atendem a(s) condição (ões) da cláusula WHERE.

Exemplos:

2.1 select NOMEA, SEXO, ANO
from TABASP
where ANO = 1 and SEXO='M';

Resultado:

NOMEA	SEXO	ANO
PAULO	M	1
SERGIO	M	1

2.2 select NOMEA, PUNIT
from TABPRODUTO
where PUNIT > 5;

Resultado:

NOMEA	PUNIT
CAFÉ	7.00
SUCO	9.00

A) SELECT - consulta a uma só tabela

3) Cláusula ORDER BY.

```
SELECT <nome dos atributos a serem mostrados>  
FROM < nome da tabela de consulta >  
ORDER BY < atributo >;
```

Funcionamento desse comando:

Monta uma tabela só com os atributos (colunas) selecionados da tabela escolhida.

A tabela resposta passa a ser ordenada (crescente) pelo atributo (coluna) escolhido.

Pode ser usado com ou sem a cláusula WHERE.

Para ordem decrescente : ORDER BY<atributo> **desc**;

Exemplos:

3.1 select NIP, NOMEA
from TABASP
order by NOMEA;

Resultado:

O NIP deixa de ordenar essa tabela.

NIP	NOMEA
4	HELENA
2	PAULA
1	PAULO
3	PEDRO
5	SERGIO

3.2 select NOMEP, PUNIT
from TABPRODUTO
where PUNIT <= 5
order by PUNIT;


Resultado:

NOMEP	PUNIT
PAO	2.00
LEITE	5.00
QUEIJO	5.00

FUNÇÕES:

Na cláusula SELECT podemos usar funções.

Principais:

- sum() - soma
 - avg() - média
 - max() - máximo
 - min() - mínimo
 - count() - conta
- usadas em atributos numéricos
-
- upper() - maiúscula
 - lower() - minúsculas
- usadas em atributos strings
- 

Exemplos de uso de Funções:

1) select avg(PUNIT)
from TABPRODUTOS;

Resultado:

avg(PUNIT)
5,6

2) select SUM(VCOMP)
from TABCOMPRA
where VCOMP > 10;

Resultado:

SUM(VCOMP)
29.00

Importante:

As tabelas geradas têm uma só linha.

O título da coluna é o nome da função. Este título ou qualquer outro pode ser alterado fazendo: `select SUM(VCOMP) as SOMA PARCIAL`.

Com o que já aprendemos, o uso de funções impede a colocação de outro atributo no `select`. Não haveria como fazer uma correlação entre colunas para formar uma tabela.

Exemplos de uso de Funções:

3) select lower(NOME)
from TABPRODUTOS;

Resultado:

NOME
pão
leite
café
suco
queijo

4) select count(*) as QUANT
from TABASP
where SEXO = 'M';

Resultado:

QUANT
3

Importante:

A função count() pode ser usada com um * no argumento, pois o que ela faz é contar quantas linhas da tabela corresponde a aspirantes do sexo Masculino.

A) SELECT - consulta a uma só tabela

4) Cláusula GROUP BY.

SELECT <FUNÇÃO, nome do atributo do group by>

FROM < nome da tabela de consulta >

GROUP BY < atributo >;

Funcionamento desse comando:

Monta uma tabela só com a coluna da função aplicada ao atributo escolhido e com o atributo do **GROUP BY**.

Pode ser usada também junto com as cláusulas WHERE e/ou ORDER BY.

Essa cláusula é usada junto com uma função no SELECT, para que essa função seja aplicada a uma **parte da tabela** e não a ela toda.

Exemplos:

```
4.1 select NIP, sum(VCOMP)
      from TACOMPRA
      group by NIP;
```

Resultado:

NIP	sum(VCOMP)
1	32,00
2	22,00
4	14,00
5	2,00

Importante:

Com o uso do **GROUP BY** foi possível colocar mais um atributo, no **SELECT**, ao lado da função.

Observe que só o atributo que está no **GROUP BY** é que pode e deve constar do **SELECT** (em qualquer ordem).

Como a soma, agora, é parcial, cabe o atributo **NIP** em conjunto com a função **SUM()**. A soma é por **NIP** e não total.

Exemplos:

```
4.2 select count(*), CODP  
      from TABCOMPRA  
      group by CODP  
      order by CODP;
```

Resultado:

count(*)	CODP
3	1
1	2
2	3
2	5

Importante:

O **ORDER BY** faz com que a tabela fique ordenada pelo **CODP**, mesmo sendo a 2ª coluna da tabela.

Como poderia ser o enunciado deste exercício?

Resposta: Indique, pelo código, em ordem crescente deste, quantas vezes um dado produto foi comprado.

B) SELECT - consulta a mais de uma tabela

Observações:

Nesses casos a cláusula WHERE passa a ser obrigatória.

Menor comando possível:

```
SELECT <nome dos atributos a serem mostrados>  
FROM < nome da tabela de consulta >  
WHERE <condições>;
```

Justificativa:

Mesmo que não haja condições específicas, coloca-se na cláusula WHERE o elo de ligação entre as tabelas consultadas.

(chave primária \leftrightarrow chave estrangeira)

Para 2 tabelas: `NIP.TABASP = NIP.TABCOMPRA` ou
`CODP.TABCOMPRA = CODP>TABPRODUTO`

Para as 3 tabelas: `NIP.TABASP = NIP.TABCOMPRA AND`
`CODP.TABCOMPRA = CODP>TABPRODUTO`