

Linguagem SQL

Comandos DML - SELECT



SELECCIONANDO REGISTROS

SELECT

```
-- COLOCANDO O BD EM USO  
USE DB_VENDAS;
```

```
-- SELECIONANDO TODAS COLUNAS  
SELECT * FROM TB_EMPREGADO;
```

```
-- SELECIONANDO ALGUMAS COLUNAS  
SELECT CODFUN, NOME, SALARIO FROM TB_EMPREGADO;
```

```
SELECT CODFUN, NOME, SALARIO, SALARIO * 1.10  
FROM TB_EMPREGADO;
```

APELIDANDO...

SELECT COM ALIAS

```
-- APELIDANDO COLUNAS
SELECT
  CODFUN as CODIGO_FUNCIONARIO,
  NOME,
  SALARIO as SALARIO_ATUAL,
  SALARIO * 1.10 as SALARIO MAIS 10 PORCENTO
FROM TB_EMPREGADO;

-- APELIDANDO COLUNAS e TABELAS
SELECT
  CODFUN Codigo, NOME "Nome",
  SALARIO 'Salário', DATA_ADMISSAO [Data de Admissão]
FROM TB_EMPREGADO;
```

afinal das CONTAS

OPERADORES MATEMÁTICOS

+ (adicionar ou concatenar)
- (subtrair)
* (multiplicar)
/ (dividir)
% (módulo – resto da divisão)

```
-- OPERADORES MATEMÁTICOS
SELECT
NOME,
SALARIO + PREMIO_MENSAL SALARIO_BONIFICADO,
SALARIO * 12 SALARIO_ANUAL,
SALARIO + PREMIO_MENSAL * 12 SALARIO_ANUAL_ERRO,
(SALARIO + PREMIO_MENSAL) * 12 SALARIO_ANUAL_OK
FROM TB_EMPREGADO
```

valores DiSTinToS

DISTINCT – VALORES DISTINTOS

```
-- VALORES DISTINTOS  
SELECT DISTINCT  
SALARIO  
FROM TB_EMPREGADO
```

```
SELECT DISTINCT  
COD_DEPTO, SALARIO  
FROM TB_EMPREGADO
```

ORDENANDO os dados

ORDER BY – ORDENANDO

```
-- ORDEM CRESCENTE  
SELECT * FROM TB_EMPREGADO  
ORDER BY NOME ASC; -- ASC É OPCIONAL
```

```
-- ORDEM DECRESCENTE  
SELECT * FROM TB_EMPREGADO  
ORDER BY SALARIO DESC; -- DESC É OBRIGATÓRIO
```

```
-- ORDEM PELA POSIÇÃO DA COLUNA  
SELECT CODFUN, NOME, SALARIO, SALARIO * 1.10 [Salário com 10% de  
aumento]  
FROM TB_EMPREGADO  
ORDER BY 3;
```

TOP ... ORDER BY ORDENANDO

```
-- ORDEM PELA POSIÇÃO DA COLUNA
SELECT CODFUN, NOME, SALARIO, SALARIO * 1.10 [Salário com 10% de aumento]
FROM TB_EMPREGADO
ORDER BY 3;

-- 10 EMPREGADO MAIS ANTIGOS
SELECT TOP 10 * FROM TB_EMPREGADO ORDER BY DATA_ADMISSAO;

-- 5 EMPREGADO QUE GANHAM MAIS
SELECT TOP 5 * FROM TB_EMPREGADO ORDER BY SALARIO DESC;

-- 5 EMPREGADO QUE GANHAM MAIS COM EMPATE
SELECT TOP 5 WITH TIES * FROM TB_EMPREGADO ORDER BY SALARIO DESC;

-- AMOSTRA DE 10% DA TABELA
SELECT TOP 10 PERCENT CODFUN, NOME, SALARIO FROM TB_EMPREGADO
ORDER BY NEWID(); --ORDENAÇÃO ALEATÓRIA
```

UÉR? FILTRANDO os dados

WHERE – FILTRANDO REGISTROS

Operadores de Comparação Escalar

=, >, >=, <, <=, <>, !=

```
-- FUNCIONÁRIOS COM SALÁRIO ABAIXO DE 1000  
SELECT CODFUN, NOME, COD_CARGO, SALARIO FROM TB_EMPREGADO  
WHERE SALARIO < 1000  
ORDER BY SALARIO;
```

```
-- FUNCIONÁRIOS COM SALÁRIO ACIMA DE 5000  
SELECT CODFUN, NOME, COD_CARGO, SALARIO FROM TB_EMPREGADO  
WHERE SALARIO > 5000  
ORDER BY SALARIO;
```

```
-- FUNCIONÁRIOS COM CAMPO COD_DEPTO IGUAL A 2  
SELECT * FROM TB_EMPREGADO  
WHERE COD_DEPTO = 2
```


WHERE – FILTRANDO REGISTROS

Operadores Lógicos

AND, OR, NOT

-- FUNCIONÁRIOS COM CAMPO COD_DEPTO IGUAL A 2 E SALARIO ACIMA DE 5000

```
SELECT * FROM TB_EMPREGADO  
WHERE COD_DEPTO = 2 AND SALARIO > 5000
```

-- FUNCIONÁRIOS COM CAMPO COD_DEPTO IGUAL A 2 OU SALARIO ACIMA DE 5000

```
SELECT * FROM TB_EMPREGADO  
WHERE COD_DEPTO = 2 OR SALARIO > 5000
```

-- FUNCIONÁRIOS COM CAMPO COD_DEPTO NÃO IGUAL A 2 AND SALARIO ACIMA DE 5000

```
SELECT * FROM TB_EMPREGADO  
WHERE NOT COD_DEPTO = 2 AND SALARIO > 5000
```

WHERE – FILTRANDO REGISTROS

Operadores Predicados

IN, BETWEEN

```
-- FUNCIONÁRIOS COM SALARIO ENTRE 3000 E 5000  
SELECT * FROM TB_EMPREGADO  
WHERE SALARIO BETWEEN 3000 AND 5000 ORDER BY SALARIO;
```

```
-- FUNCIONÁRIOS COM SALARIO MENOR QUE 3000 OU MAIOR QUE 5000  
SELECT * FROM TB_EMPREGADO  
WHERE SALARIO NOT BETWEEN 3000 AND 5000 ORDER BY SALARIO;
```

```
-- FUNCIONÁRIOS SÃO DOS DEPARTAMENTOS 2,3,4  
SELECT * FROM TB_EMPREGADO WHERE COD_DEPTO IN (2,3,4)
```

```
-- FUNCIONÁRIOS QUE NÃO SÃO DOS DEPARTAMENTOS 2,3,4  
SELECT * FROM TB_EMPREGADO WHERE COD_DEPTO NOT IN (2,3,4)
```


WHERE – FILTRANDO REGISTROS

Caracteres Coringas

%, _, []

```
-- EMPREGADOS COM NOME MARIA  
SELECT * FROM TB_EMPREGADO  
WHERE NOME = 'MARIA';
```

```
-- NOMES QUE COMEÇAM COM MARIA  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE 'MARIA%';
```

```
-- NOMES QUE TERMINAM COM MARIA  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '%MARIA';
```

WHERE – FILTRANDO REGISTROS

Caracteres Coringas

%, _, []

```
-- NOMES QUE CONTENHAM SOUZA  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '%SOUZA%';
```

```
-- NOMES QUE CONTENHAM SOUZA OU SOUSA  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '%SOU[SZ]A%'; -- NOME LIKE '%SOUZA%' OR NOME LIKE  
'%SOUSA%'
```

```
-- INICIADOS POR QUALQUER CARACTERE E SEGUNDO CARACTERE A  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '_A%';
```

WHERE – FILTRANDO REGISTROS

Caracteres Coringas

%, _, []

```
-- NOMES CUJO PENÚLTIMO CARACTERE SEJA A LETRA Z  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '%Z_';
```

```
-- NOMES TERMINADOS EM LU E SEGUIDOS DE 3 OUTRAS LETRAS  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '%LU___';
```

```
-- NOMES QUE CONTENHAM PRIMEIRA LETRA SEJA M OU P E TERMINEM COM A  
OU O  
SELECT * FROM TB_EMPREGADO  
WHERE NOME LIKE '[MP]%[AO]'
```

WHERE – FILTRANDO REGISTROS

Valores Nulos

is null, is not null

-- EMPREGADO SEM SALÁRIO

```
SELECT CODFUN, NOME, SALARIO, PREMIO_MENSAL  
FROM TB_EMPREGADO WHERE SALARIO IS NULL;
```

-- EMPREGADO COM SALÁRIO

```
SELECT CODFUN, NOME, SALARIO, PREMIO_MENSAL  
FROM TB_EMPREGADO WHERE SALARIO IS NOT NULL;
```

-- TRATANDO NULOS

```
SELECT CODCLI, NOME, ISNULL(ESTADO, 'NÃO INFORMADO') ESTADO TRATADO,  
COALESCE(EST_COB, ESTADO, 'NC') AS EST_COBRANCA, EST_COB, ESTADO  
FROM CLIENTES ORDER BY 3;
```

Mão na massa – Individual



Lista de Exercícios – SELECT

1. Coloque em uso o banco de dados DB_MADA_VAREJO.
2. Selecione todas as colunas de todas as tabelas (um select por tabela).
3. Selecione as colunas nome e email de todos os clientes.
4. Exiba todos os produtos com seu nome, código (não o utilize o ID), preço de venda, preço de custo e calcule o lucro estimado de cada produto. Ordene os produtos pelo valor do lucro estimado, trazendo primeiro os que terão maior lucro.
5. Exiba quais as cidades de São Paulo cadastradas nos endereços. Se a cidade possuir mais de um endereço, exibir uma única vez.
6. Exibir todos os dados das 5 vendas mais recentes que estão em transporte.
7. Retornar os produtos com custo superior a R\$1.000 que começaram a ser vendidos em 2021.
8. Exibir o nome das categorias que não possuem categorias superiores.
9. Exibir todos os dados dos produtos da marca multilaser
10. Selecionar as vendas de 2019 que foram canceladas e que tinham frete grátis.
11. Exibir os dados de todos os endereços que são avenidas do Sudeste

Obrigado.

Colabore com sua opinião!



<https://www.menti.com/353szq8333>

