

```

use db_Pedidos

/* configurar o Studio para:
1) mensagens de erros
2) formatações para data(dd/mm/yyyy)
3) valores monetários R$ 0.000,00

Definição: Microsoft
Especifica o ambiente de idioma para a sessão.
O idioma da sessão determina os formatos de datetime e as mensagens do sistema.
*/
set language brazilian

/* 1) retornar os produtos
usa-se colchetes[] para apelidos com
espaços entre palavras
Link referente as formatações
https://learn.microsoft.com/pt-br/sql/t-sql/functions/format-transact-sql?view=sql-server-ver17

1) No Select, retorne os campos: ID_PRODUTO, DESCRICAO, PRECO_CUSTO,
PRECO_VENDA
2) tabela: TB_PRODUTO
3) formate em moeda R$ os campos: PRECO_CUSTO e PRECO_VENDA
*/
select
    ID_PRODUTO as CODIGO,
    DESCRICAO 'DESCRICAO DO PRODUTO',
    -- formatação contábil
    format(PRECO_CUSTO, 'C', 'PT-BR') as [PRECO CUSTO],
    -- formatação contábil
    format(PRECO_VENDA, 'C', 'PT-BR') as [PRECO VENDA]
from TB_PRODUTO
-----  

-----  

/* 2) mostrar os pedidos
e formatar a coluna DATA_EMISSAO para dd/MM/yyyy
Tabela: TB_Pedido

"mm" minúsculo significa um minuto em uma hora, mas "MM"
maiúsculo significa o mês do ano. Eles não devem ser misturados.
Os meses devem ser usados junto com o ano ou dia, enquanto os
minutos devem ser usados junto com uma hora.
*/
select
    NUM_PEDIDO as [Nº PEDIDO],
    FORMAT(DATA_EMISSAO, 'dd/MM/yyyy') as [DATA FORMATADA],
    FORMAT(VLR_TOTAL, 'C', 'PT-BR') as [R$ TOTAL],
    SITUACAO,
    OBSERVACOES
from TB_Pedido
-----  

-----  

-- 3) selecionar os 10 primeiros registros

```

```

-- Tabela: TB_Pedido
select
    top 10 *
from TB_PEDIDO
order by NUM_PEDIDO
-----
-----
-- 4) selecionar os 15 últimos registros e
-- classificar em ordem do maior para o menor [Z-A]
-- Tabela: TB_Pedido
SELECT TOP 15 *
FROM TB_PEDIDO
ORDER BY NUM_PEDIDO DESC
-----
-----
-- 5) Exibir da linha 100 até 300
--      OFFSET 99 ROWS pula os 99 primeiros registros (ou seja, começa no 100).
--      FETCH NEXT 201 ROWS ONLY pega os 201 registros seguintes, ou seja, do 100
até o 300.
--      Importante: você precisa usar o ORDER BY, senão o SQL Server não aceita
OFFSET-FETCH.

-- Definições finais:
--      "OFFSET" em inglês significa "deslocamento" ou "salto"
--      "ROWS" linhas
--      "FETCH NEXT" é usado para trazer um número específico de linhas a
partir da posição indicada pelo OFFSET.
SELECT
    NUM_PEDIDO,
    CODCLI,
    CODVEN,
    DATA_EMISSAO,
    VLR_TOTAL
FROM TB_PEDIDO
ORDER BY NUM_PEDIDO
OFFSET 99 ROWS FETCH NEXT 201 ROWS ONLY
-----
-----
-- 6) selecionar os 10% dos primeiros registros
-- TB_CLIENTE
select
    top(25) percent *
from TB_CLIENTE
order by codcli desc
-----
-----
-- 7) listar as 7 pessoas com o maior salário
-- TB_EMPREGADO
select
    top(7) *
from TB_EMPREGADO
order by SALARIO desc

```

```
-----  
-- 8) Será que a análise anterior está correta ?  
-- Mas realmente quais são as pessoas com o maior  
-- salário inclusive comparadas ao último ?  
  
-- WITH TIES é usado para incluir linhas que correspondam aos valores da última  
linha  
-- TB_EMPREGADO  
select  
    top(7) with ties *  
from TB_EMPREGADO  
order by SALARIO desc
```

```
-- utilização do comando  
-- SELECT com clausula WHERE(filtro)  
use db_Pedidos  
  
/* configurar o Studio para:  
1) mensagens de erros  
2) formatações para data(dd/mm/yyyy)  
3) valores monetários R$ 0.000,00
```

```
Definição: Microsoft  
Especifica o ambiente de idioma para a sessão.  
O idioma da sessão determina os formatos de datetime e as mensagens do sistema.  
*/  
set language brazilian
```

```
-- 1) tabela: TB_Cliente  
-- filtrar os registros onde o Estado = SP  
select  
    NOME,  
    FANTASIA,  
    CIDADE,  
    ESTADO -- campos da tabela  
from TB_CLIENTE -- tabela origem(onde os dados estão)  
where ESTADO = 'SP' -- filtrar pela coluna
```

```
/* 2) tabela: TB_Cliente  
filtrar os registros por  
Estado = SP  
*/  
select  
    NOME,  
    FANTASIA,  
    E_MAIL,  
    ICMS  
from TB_CLIENTE
```

```
where ESTADO='SP'
```

```
/* 3) tabela: TB_Cliente  
filtrar os registros na coluna Nome que começam com JOSE ou Maria--COMANDO LIKE  
select  
    NOME,  
    CNPJ,  
    ICMS  
from TB_CLIENTE  
where NOME LIKE 'JOSE%' or NOME LIKE 'MARIA%'
```

```
/* 4) tabela: TB_CLIENTE  
retornar somente os registros  
onde a coluna Nome contém Sousa ou Souzaselect  
    CODCLI,  
    NOME,  
    CIDADE,  
    ESTADO,  
    E_MAIL  
from TB_CLIENTE  
where NOME LIKE '%SOU[SZ]A%'
```

```
/* 5) tabela: TB_CLIENTE  
retornar todos os registros  
onde a coluna Nome contenha LTDASELECT  
    NOME,  
    CIDADE,  
    ESTADO,  
    E_MAIL  
FROM TB_CLIENTE  
WHERE NOME LIKE '%LTDA%';
```

```

-- clausula where filtros com Dataas

/* 6) tabela: TB_EMPREGADO
filtrar DATA_ADMISSAO >= '01-01-2000' AND DATA_ADMISSAO <='31-12-2000'
*/
select
    NOME,
    format(DATA_ADMISSAO, 'dd/MM/yyyy') as ADMISSAO,
    format(SALARIO, 'c', 'pt-br') as SALARIO,
    format(PREMIO_MENSAL, 'c', 'pt-br') as [PREMIO MENSAL]
from TB_EMPREGADO
where DATA_ADMISSAO >= '01-01-2000' AND DATA_ADMISSAO <='31-12-2000'

-----
-----
```

```

-- 7)retornar os registros para o ANO específico ( É MAIS RAPIDO )
-- tabela: TB_EMPREGADO
select NOME,
       format(DATA_ADMISSAO, 'dd/MM/yyyy') as ADMISSÃO
from TB_EMPREGADO
where YEAR(DATA_ADMISSAO) = 2000

-----
```

```

-- 8) tabela: TB_EMPREGADO
-- retornar os registros para os anos da LISTA (IN)
select
    NOME,
    SALARIO,
    format(DATA_ADMISSAO, 'dd/MM/yyyy') as ADMISSAO
from TB_EMPREGADO
where YEAR(DATA_ADMISSAO) IN (2000, 2005, 2010)

-----
```

```

/* 9)
tabela: TB_EMPREGADO
retornar registros agrupados pela coluna: DATA_ADMISSAO
e somar a coluna SALARIO
```

```

*/
select
    year(DATA_ADMISSAO) as ADMISSAO,
    SALARIO as Total_Formatado
from TB_EMPREGADO
order by year(DATA_ADMISSAO) asc

----- agrupado fica neste modelo ( MELHOR E ENXUTO )
select
    year(DATA_ADMISSAO) as ADMISSAO,
    format(avg(SALARIO), 'R$ ###,##0.00') as Total_Formatado
from TB_EMPREGADO
group by year(DATA_ADMISSAO)
order by year(DATA_ADMISSAO) asc
```

```

10) tabela: TB_CLIENTE, retornar os registros da região SUDESTE
select
    NOME,
    ENDERECO,
    CIDADE,
    ESTADO
from TB_CLIENTE
where ESTADO in ('SP', 'RJ', 'ES', 'MG')
order by ESTADO asc -- ou desc

-----
-- 11) tabela: TB_CLIENTE
-- retornar os estados diferentes de: ('SP', 'RJ', 'ES', 'MG')
select
    NOME,
    ENDERECO,
    CIDADE,
    ESTADO
from TB_CLIENTE
where ESTADO NOT in ('SP', 'RJ', 'ES', 'MG')
order by ESTADO

```

Exercícios:

- A) ATENÇÃO COLOQUE O NOME DO BANCO DE DADOS ***
`use db_Pedidos`

-- B) Coloque o código para formatação no padrão brasileiro
`set language brazilian`
-- Exercícios com SELECT

-- 1) retorne as colunas: NOME, FANTASIA, CIDADE
-- origem: TB_CLIENTE
-- 562 registros
select
 NOME,
 FANTASIA,
 CIDADE
from TB_CLIENTE

-- 2) retorne as colunas: NOME, FANTASIA, CIDADE
-- origem: TB_CLIENTE
-- filtro: CIDADE = Rio de Janeiro
-- 21 registros
select
 NOME,

```
FANTASIA,
CIDADE
from TB_CLIENTE
where CIDADE = 'Rio de Janeiro'

-----
-- 3) retorne as colunas: FANTASIA, ESTADO
-- origem: TB_CLIENTE
-- filtre: ESTADO = 'MINAS GERAIS'
-- 42 registros
select
    FANTASIA,
    ESTADO
from TB_CLIENTE
where ESTADO = 'MG'

-----
-- 4) retorne as colunas: BAIRRO, NOME, FANTASIA
-- origem: TB_CLIENTE
-- filtre os BAIRROS = BRÁS, CASA VERDE, CENTRO, IPIRANGA
-- classifique BAIRRO em ordem alfabética
-- 97 registros
select
    BAIRRO,
    NOME,
    FANTASIA
from TB_CLIENTE
where BAIRRO in ('BRAS', 'CASA VERDE', 'CENTRO', 'IPIRANGA')
order by BAIRRO

-----
-- 5) retorne as colunas: NOME, SALARIO, PREMIO_MENSAL
-- origem: TB_EMPREGADO
-- filtre o NOME que contém OLIVEIRA
-- classifique NOME em ordem alfabética
-- 8 registros retornados
select
    NOME,
    SALARIO,
    PREMIO_MENSAL
from TB_EMPREGADO
where NOME like '%OLIVEIRA%'
order by NOME
```

```

-----
-- 6) retorne as colunas: NOME, SALARIO, PREMIO_MENSAL
-- origem: TB_EMPREGADO
-- filtre o SALARIO maior do que três mil
-- classifique SALARIO do maior para o menor
-- formate as colunas SALARIO e PREMIO_MENSAL para o padrão brasileiro(contábil
R$)
-- 22 registros retornados
select
    NOME,
    format(SALARIO, 'c', 'pt-br') as [SALARIO],
    format(PREMIO_MENSAL, '') as [PRÊMIO MENSAL]
from TB_EMPREGADO
where SALARIO >= 3000
order by SALARIO desc
-----

-- 7) retorne as colunas: NOME, SALARIO, PREMIO_MENSAL
-- origem: TB_EMPREGADO
-- filtre o PREMIO_MENSAL menor do que mil reais
-- classifique PREMIO_MENSAL do maior para o menor
-- formate as colunas SALARIO e PREMIO_MENSAL para o padrão brasileiro(contábil
R$)
-- 36 registros
select
    NOME,
    format(SALARIO, 'c', 'pt-br') as [SALARIO],
    format(PREMIO_MENSAL, '') as [PRÊMIO MENSAL]
from TB_EMPREGADO
where PREMIO_MENSAL < 1000
order by PREMIO_MENSAL desc

```

Exercício extra:

```

use db_Pedidos

set language brazilian

-- 8) retorne as colunas COD FORNECEDOR, NOME, NOME FANTASIA,
FISICA_JURIDICA, BAIRRO
-- origem: TB FORNECEDOR
-- filtre: NOME com início 'ACO' e BAIRRO igual 'ACLIMAÇÃO'
-- 1 registro retornado
select
    COD_FORNECEDOR,
    NOME,
    NOME_FANTASIA,
    FISICA_JURIDICA,
    BAIRRO
from TB_FORNECEDOR
WHERE NOME like 'ACO%' and BAIRRO = 'ACLIMACAO'

```

```
-- 9) retorne as colunas: COD FORNECEDOR, NOME, NOME FANTASIA,  
FISICA_JURIDICA, BAIRRO  
-- origem: TB FORNECEDOR  
-- filtre: NOME com inicio 'ACO' e BAIRRO igual 'móoca'  
-- 3 registros retornados  
select  
    COD_FORNECEDOR,  
    NOME,  
    NOME_FANTASIA,  
    FISICA_JURIDICA,  
    BAIRRO  
from TB_FORNECEDOR  
WHERE NOME like 'ACO%' and BAIRRO = 'MOOCA'  
-----  
-----  
-- 10) retorne as colunas: NOME, DATA ADMISSAO, SALARIO, SINDICALIZADO  
-- origem: TB EMPREGADO  
-- filtre: SINDICALIZADO igual a sim e o ano da DATA ADMISSAO igual à 1986  
-- classificado por salário crescente  
-- 5 linhas retornadas  
select  
    NOME,  
    DATA_ADMISSAO,  
    SALARIO,  
    SINDICALIZADO  
from TB_EMPREGADO  
where SINDICALIZADO = 'S' and  
      year(DATA_ADMISSAO) = 1986
```