

INNER JOIN

```
use db_Pedidos
set language brazilian

/*
1) relacionar: TB_CLIENTE x TB_PEDIDO
a) retornar: TB_CLIENTE = COD_CLI, NOME
b) retornar: TB_PEDIDO = NUM_PEDIDO, VLR_TOTAL
c) classificar por CODCLI crescente
*/
select
    CLIENTE.CODCLI,
    CLIENTE.NOME,
    PEDIDO.NUM_PEDIDO,
    PEDIDO.VLR_TOTAL
-- Origem TB_CLIENTE apelido(CLIENTE) unir
-- com TB_PEDIDO apelido(PEDIDO)
from TB_CLIENTE CLIENTE
    join TB_PEDIDO PEDIDO
        on CLIENTE.CODCLI = PEDIDO.CODCLI
order by CLIENTE.CODCLI
-----

/*
2) relacionar: TB_CLIENTE x TB_PEDIDO x TB_VENDEDOR
a) retornar: TB_CLIENTE = CODCLI, NOME
b) retornar: TB_PEDIDO = NUM_PEDIDO, VLR_TOTAL
c) retornar: TB_VENDEDOR = NOME
d) exibir os clientes e seus pedidos onde o vendedor é o Gilson
*/
select
    CLIENTE.CODCLI,
    CLIENTE.NOME,
    PEDIDO.NUM_PEDIDO,
    PEDIDO.VLR_TOTAL,
    VENDEDOR.NOME as [NOME VENDEDOR]
from TB_CLIENTE CLIENTE
    join TB_PEDIDO PEDIDO
        on CLIENTE.CODCLI = PEDIDO.CODCLI
    join TB_VENDEDOR VENDEDOR
        on VENDEDOR.CODVEN = PEDIDO.CODVEN
where VENDEDOR.NOME = 'GILSON'
```

```

-----
/*
3) relacionar: TB_VENDEDOR x TB_PEDIDO
a) retornar: TB_VENDEDOR = CODVEN, NOME, VLR_TOTAL
b) formatar VLR_TOTAL contábil
c) mostre o [ TOTAL DE VENDAS ] por cada vendedor
d) arredonde o [ TOTAL DE VENDAS ] sem casas decimais
e) classifique pelo NOME do vendedor
*/
select
    VENDEDOR.CODVEN,
    VENDEDOR.NOME as [NOME VENDEDOR],
    format(round(sum (PEDIDO.VLR_TOTAL),0), '###,##0.') [TOTAL VENDIDO]
from TB_PEDIDO PEDIDO
    join TB_VENDEDOR VENDEDOR
        on VENDEDOR.CODVEN = PEDIDO.CODVEN
group by VENDEDOR.CODVEN, VENDEDOR.NOME
order by VENDEDOR.NOME

```

```

-----
/*
4) relacionar: TB_PEDIDO x TB_CLIENTE
a) retornar: TB_PEDIDO = NUM_PEDIDO, DATA_EMISSAO
b) retornar: TB_CLIENTE = CODCLI, NOME, CIDADE, ESTADO
c) mostre os clientes da cidade Jundiaí do estado SP
    somente para o ano 2014
d) formatar coluna DATA_EMISSAO = dd/MM/yyyy
e) classificar DATA_EMISSAO crescente
*/
select
    PEDIDO.NUM_PEDIDO,
    CLIENTE.CODCLI,
    format(PEDIDO.DATA_EMISSAO, 'dd/MM/yyyy') [DATA EMISSAO],
    CLIENTE.NOME,
    CLIENTE.CIDADE,
    CLIENTE.ESTADO
from TB_PEDIDO PEDIDO
    join TB_CLIENTE CLIENTE
        on PEDIDO.CODCLI = CLIENTE.CODCLI
where CLIENTE.CIDADE = 'JUNDIAI' and
    CLIENTE.ESTADO = 'SP' and
    year(PEDIDO.DATA_EMISSAO) = 2014
order by PEDIDO.DATA_EMISSAO

```

```

/*
5) TB_EMPREGADO X TB_CARGO
a) retornar: TB_EMPREGADO = NOME, DATA_ADMISSÃO, SALARIO
b) retornar: TB_CAGO = CARGO
c) mostre os colaboradores e seus cargos
*/
select
    EMPREGADO.NOME,
    EMPREGADO.DATA_ADMISSAO,
    EMPREGADO.SALARIO,
    CARGO.CARGO
from TB_EMPREGADO EMPREGADO
    join TB_CARGO CARGO
        on EMPREGADO.COD_CARGO = CARGO.COD_CARGO
order by EMPREGADO.NOME

```

```

-----
/*
6) TB_EMPREGADO X TB_CARGO X TB_DEPARTAMENTO
a) retornar: TB_EMPREGADO = NOME, DATA_ADMISSÃO, SALARIO
b) retornar: TB_CAGO = CARGO
c) retornar: TB_DEPARTAMENTO = DEPTO
c) mostre: os colaboradores com seus cargos e seus departamentos
*/
select
    EMPREGADO.NOME,
    EMPREGADO.DATA_ADMISSAO,
    EMPREGADO.SALARIO,
    CARGO.CARGO,
    DEPARTAMENTO.DEPTO
from TB_EMPREGADO EMPREGADO
    join TB_CARGO CARGO
        on EMPREGADO.COD_CARGO = CARGO.COD_CARGO
    join TB_DEPARTAMENTO DEPARTAMENTO
        on DEPARTAMENTO.COD_DEPTO = EMPREGADO.COD_DEPTO
order by EMPREGADO.NOME

```

LEFT JOIN e RIGHT JOIN.

```

use db_Pedidos
set language brazilian

-- 1) TB_DEPARTAMENTO x TB_EMPREGADO
--    === left ===

-- retornar TB_DEPARTAMENTO = COD_DEPTO, DEPTO
-- retornar TB_EMPREGADO = NOME
-- Faça análise: Quais departamentos estão sem empregados ?

select
    DEPARTAMENTO.COD_DEPTO,
    DEPARTAMENTO.DEPTO,
    EMPREGADO.NOME,
    format(EMPREGADO.SALARIO, 'C', 'PT-BR') As SALÁRIO
from TB_DEPARTAMENTO DEPARTAMENTO
    left join TB_EMPREGADO EMPREGADO
        on DEPARTAMENTO.COD_DEPTO = EMPREGADO.COD_DEPTO

```

```
order by EMPREGADO.NOME
```

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

```
/* 2)
```

```
-- Faça análise:
```

```
Quais clientes fizeram pedidos ?
```

```
OBS.: o CLIENTE 'BBBB' foi cadastrador porém houve erro no sistema ao registrar  
no banco de dados
```

```
*/
```

```
select
```

```
    cliente.NOME,  
    ped.NUM_PEDIDO,  
    prod.DESCRICAO,  
    itens.QUANTIDADE,  
    itens.PR_UNITARIO,  
    ped.VLR_TOTAL
```

```
from TB_CLIENTE cliente
```

```
    left join TB_PEDIDO ped
```

```
        on cliente.CODCLI = ped.CODCLI
```

```
    left join TB_ITENSPEDIDO itens
```

```
        on itens.NUM_PEDIDO = ped.NUM_PEDIDO
```

```
    left join TB_PRODUTO prod
```

```
        on prod.ID_PRODUTO = itens.ID_PRODUTO
```

```
order by ped.NUM_PEDIDO
```

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

```
/* 3)
```

```
-- TB_DEPARTAMENTO x TB_EMPREGADO
```

```
--          === right ===
```

```
faça análise: Quais empregados estão sem departamentos ?
```

```
-- retorne: NOME, DEPTO
```

```
*/
```

```
select
```

```
    EMPREGADO.NOME,  
    DEPARTAMENTO.DEPTO
```

```
from
```

```
    TB_DEPARTAMENTO DEPARTAMENTO
```

```
        right outer join TB_EMPREGADO EMPREGADO
```

```
            on EMPREGADO.COD_DEPTO = DEPARTAMENTO.COD_DEPTO
```

```
order by DEPARTAMENTO.DEPTO
```

```

-----
/*
7) TB_DEPARTAMENTO x TB_EMPREGADO
a) retornar: TB_DEPARTAMENTO = DEPTO
b) faça uma contagem do departamento
c) mostre [ QTDE COLABORADOR ] por departamento
*/

-- neste resultado temos: DEPTO e NOME

select
    DEPARTAMENTO.DEPTO,
    EMPREGADO.NOME
from TB_DEPARTAMENTO DEPARTAMENTO
    join TB_EMPREGADO EMPREGADO
    on DEPARTAMENTO.COD_DEPTO = EMPREGADO.COD_DEPTO
order by DEPARTAMENTO.DEPTO

-- nesta fase faremos a contagem inicial solicitada
select
    DEPARTAMENTO.DEPTO,
    count(DEPARTAMENTO.DEPTO) [QTDE COLABORADOR]
from TB_DEPARTAMENTO DEPARTAMENTO
    join TB_EMPREGADO EMPREGADO
    on DEPARTAMENTO.COD_DEPTO = EMPREGADO.COD_DEPTO
group by DEPARTAMENTO.DEPTO
order by DEPARTAMENTO.DEPTO

```

SELECTS

```

use db_Pedidos;

set language brazilian;

-- TB_CLIENTE
-- 1) retorne somente a coluna: Cidade
Select
    CIDADE
From TB_CLIENTE
Order By CIDADE;

-- 2) TB_CLIENTE
-- a) retorne as CIDADES SEM DUPLICIDADE
Select
    Distinct -- remove as duplicidades
    CIDADE
From TB_CLIENTE
Order By CIDADE;

```

```
-- 3)
-- RTRIM() = Remove os espaços à direita do texto
-- LTRIM() = Remove os espaços à esquerda do texto
-- TRIM() = ** é o melhor ** Remove os espaços à esquerda e à direita do texto
```

```
Select
    Distinct
    Trim(CIDADE)
From TB_CLIENTE
Order By Trim(CIDADE);
```

```
/*
4)
```

Funções: LEN, SUBSTRING

LEN: a função LEN() retorna o comprimento(tamanho) de um texto, conta a quantidade de "dígitos/caracteres" do texto, no Excel equivale a função =Núm.Caract()

SUBSTRING: a função SUBSTRING() extrai alguns caracteres de um texto, seu principal objetivo é cortar o TEXTO de acordo com a posição no Excel equivale a função =Ext.Texto()

```
*/
Select
    NOME,
    LEN(CEP_COB) as [QTDE CARACTERES],
    CEP_COB,
    SUBSTRING(CEP_COB,1,5) + '-' + SUBSTRING(CEP_COB,6,3) as CEP
From TB_CLIENTE
Where LEN(TRIM(CEP_COB)) = 8
```