

```

1  /*
2  Banco de Dados
3  Prof.ª Mariana Meirelles de Mello
4  Semestre: 2025.2
5  Data: 25/11/2025
6
7  Aula 06: Consulta em mais de uma tabela
8 */
9
10 -- 7. Mostre o nome dos moradores e o código do bairro (fk_idBairro), mas usando
11 alias: nome AS Morador, fk_idBairro AS Código_Bairro.
12 SELECT nome AS Morador, fk_idBairro AS Código_Bairro
13 FROM morador;
14
15 SELECT idBairro, nome
16 FROM bairro;
17
18 /*
19 Não devemos realizar SELECTs separados para buscar as informações que desejamos.
20 Podemos unir as tabelas através do uso do JOIN. Temos diferentes tipos de JOIN:
21 - INNER JOIN: retorna as linhas que tem valores correspondentes em todas as tabelas.
22 - LEFT JOIN: retorna todas as linhas da tabela a esquerda (a que aparecer antes no
JOIN), mesmo que não tenha correspondência com as outras tabelas.
23 - RIGHT JOIN: retorna todas as linhas da tabela a direita (a que aparecer por último
no JOIN), mesmo que não tenha correspondência com as outras tabelas.
24
25 Sintaxe:
26 SELECT campo1, campo2, campoN
27 FROM tabela1 (alias)
28     INNER/LEFT/RIGHT JOIN tabela2 (alias)
29     ON chave_estrangeira = chave_primaria;
30
31 O 'alias' é opcional, mas é muito utilizado para tirar duplicidade de nomes de campos
iguais das tabelas. Verifique o uso do 'alias' no exemplo abaixo.
32
33 No caso da consulta solicitada vamos usar um INNER JOIN, pois queremos que mostre os
professores e suas respectivas disciplinas. Os professores que não tem disciplina
associada, ou vice-versa, não nos interessa.
34 */
35
36 USE aula01_noite;
37
38 SHOW TABLES;
39
40 DESC cliente;
41 DESC pedido;
42
43 SELECT * FROM cliente;
44 SELECT * FROM pedido;
45
46 -- Mostra os clientes que tem pedidos
47 SELECT nome
48 FROM cliente
49     INNER JOIN pedido
50     ON id = fk_idcliente;
51
52 -- Mostre todos os clientes, mesmo aqueles que não fizeram pedidos
53 SELECT nome, valor
54 FROM cliente
55     LEFT JOIN pedido
56     ON id = fk_idcliente;
57
58 SELECT nome, valor
59 FROM pedido
60     RIGHT JOIN cliente
61     ON id = fk_idcliente;
62
63 -- Mostre todos os pedidos, mesmo sem cliente
64 SELECT valor, nome
65 FROM cliente
66     RIGHT JOIN pedido
67     ON id = fk_idcliente;

```

```
68 USE sistema_comunitario;
69
70 /*
71 Vamos voltar ao exemplo pedido em aulas anteriores.
72
73 -- 7. Mostre o nome dos moradores e o código do bairro (fk_idBairro), mas usando
74 alias: nome AS Morador, fk_idBairro AS Codigo_Bairro.
75 */
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
```

USE sistema\_comunitario;

/\*

Vamos voltar ao exemplo pedido em aulas anteriores.

-- 7. Mostre o nome dos moradores e o código do bairro (fk\_idBairro), mas usando alias: nome AS Morador, fk\_idBairro AS Codigo\_Bairro.

\*/

**SELECT** m.nome 'Morador', b.nome 'Bairro'

**FROM** morador m

**INNER JOIN** bairro b

**ON** m.fk\_idbairro = b.idBairro

**ORDER BY** m.nome;

-- Mostre os eventos e seus bairros

**SELECT** e.titulo 'Evento', b.nome 'Bairro'

**FROM** evento e

**INNER JOIN** bairro b

**ON** e.fk\_idBairro = b.idBairro

**ORDER BY** e.titulo;

-- Mostre todos os moradores e os seus eventos, mesmo aqueles que não tenham organizado evento

**SELECT** m.nome 'Morador', e.titulo 'Evento'

**FROM** morador m

**LEFT JOIN** evento e

**ON** m.idMorador = e.fk\_idMorador

**WHERE** e.fk\_idMorador **IS NULL**;

/\*DESAFIO:

Mostre o nome e o telefone dos moradores do bairro "Mangabeira".

\*/

**SELECT** m.nome, m.telefone

**FROM** morador m

**INNER JOIN** bairro b

**ON** b.idbairro = m.fk\_idBairro

**WHERE** b.nome = 'Mangabeira';