

```

1  /*
2  Banco de Dados
3  Prof.ª M.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 Codigo_Bairro.
12 SELECT nome AS Morador, fk_idbairro AS Codigo_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
23 JOIN), mesmo que não tenha correspondência com as outras tabelas.
24 - RIGHT JOIN: retorna todas as linhas da tabela a direita (a que aparecer por último
25 no JOIN), mesmo que não tenha correspondência com as outras tabelas.
26
27 Sintaxe:
28 SELECT campo1, campo2, campoN
29 FROM tabela1 (alias)
30     INNER/LEFT/RIGHT JOIN tabela2 (alias)
31     ON chave_estrangeira = chave_primaria;
32
33 O 'alias' é opcional, mas é muito utilizado para tirar duplicidade de nomes de campos
34 iguais das tabelas. Verifique o uso do 'alias' no exemplo abaixo.
35
36 No caso da consulta solicitada vamos usar um INNER JOIN, pois queremos que mostre os
37 professores e suas respectivas disciplinas. Os professores que não tem disciplina
38 associada, ou vice-versa, não nos interessa.
39 */
40
41 USE aula01_noite;
42
43 SHOW TABLES;
44
45 DESC cliente;
46 DESC pedido;
47
48 SELECT * FROM cliente;
49 SELECT * FROM pedido;
50
51 -- Mostra os clientes que tem pedidos
52 SELECT nome
53 FROM cliente
54     INNER JOIN pedido
55     ON id = fk_idcliente;
56
57 -- Mostre todos os clientes, mesmo aqueles que não fizeram pedidos
58 SELECT nome, valor
59 FROM cliente
60     LEFT JOIN pedido
61     ON id = fk_idcliente;
62
63 SELECT nome, valor
64 FROM pedido
65     RIGHT JOIN cliente
66     ON id = fk_idcliente;
67
68 -- Mostre todos os pedidos, mesmo sem cliente
69 SELECT valor, nome
70 FROM cliente
71     RIGHT JOIN pedido
72     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  SELECT m.nome 'Morador', b.nome 'Bairro'
77  FROM morador m
78       INNER JOIN bairro b
79       ON m.fk_idbairro = b.idBairro
80  ORDER BY m.nome;
81
82  -- Mostre os eventos e seus bairros
83  SELECT e.titulo 'Evento', b.nome 'Bairro'
84  FROM evento e
85       INNER JOIN bairro b
86       ON e.fk_idBairro = b.idBairro
87  ORDER BY e.titulo;
88
89  -- Mostre todos os moradores e os seus eventos, mesmo aqueles que não tenham
90  organizado evento
91  SELECT m.nome 'Morador', e.titulo 'Evento'
92  FROM morador m
93       LEFT JOIN evento e
94       ON m.idMorador = e.fk_idMorador
95  WHERE e.fk_idMorador IS NULL;
96
97  /*DESAFIO:
98  Mostre o nome e o telefone dos moradores do bairro "Mangabeira".
99  */
100 SELECT m.nome, m.telefone
101 FROM morador m
102       INNER JOIN bairro b
103       ON b.idbairro = m.fk_idBairro
104 WHERE b.nome = 'Mangabeira';

```