

INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA DE SOFTWARE



Lista de exercícios sobre JOIN

20 exercícios

Prof: Leonardo Glória

Aluno: Richard de Jesus Cabral Alves.

1- Utilize os dados que estão nesse GITHUB:

[\[https://github.com/leoinfnet/lista_exercicio_joins_sql_2025\]](https://github.com/leoinfnet/lista_exercicio_joins_sql_2025)[dados!]

2- Voce deverá criar o container, criar o schema, e database criar as tabelas e inserir os dados.

Exercícios 1–20

1 - (pedidos × clientes).

Liste [pedido.id](#), cliente.nome, pedido.criado_em. Ordene por pedido.criado_em desc.

2 -com cupom opcional.

Traga [pedido.id](#), COALESCE(cupons.codigo, 'SEM_CUPOM') AS cupom, COALESCE(cupons.percentual, 0) AS desconto_percentual.

3 - : vendedores sem livros.

Liste todos os vendedores com COUNT([livros.id](#)) (pode ser 0). Mostre [vendedor.id](#), vendedor.nome_fantasia, qtde_livros. Ordene por qtde_livros asc.

4- (categorias pai/filha).

Mostre categoria_filha.nome AS filha, categoria_pai.nome AS pai. Inclua categorias raiz (sem pai) usando LEFT JOIN em si mesma.

5 - (grade de horários).

Gere combinações dias × horas. Filtre apenas horários de 09:00 a 18:00. Mostre dia, hora.
(Se não criar tabelas dias/horas, pode usar CTEs para gerar séries.)

6 - JOIN em cadeia (pedidos → itens → livros).

Mostre [pedido.id](#), livro.titulo, itens.quantidade, itens.preco_unitario, ROUND(itens.quantidade * itens.preco_unitario, 2) AS subtotal.

7 - CASE por faixa de preço.

Liste livros com CASE que classifica preco: 'barato' (<30), 'medio' (30–80),

'caro' (>80). Inclua categoria.nome via LEFT JOIN.

8 - Cidades com e sem clientes

Traga cidades.nome, COUNT([clientes.id](#)) AS qtde_clientes, incluindo cidades sem clientes. Ordene por qtde_clientes desc.

9 - Faturamento por vendedor no mês .

Para o mês corrente (DATE_TRUNC('month', pedidos.criado_em)), some itens.quantidade * itens.preco_unitario por vendedor. Inclua vendedores sem vendas no mês com COALESCE(total, 0).

10 - Livros nunca vendidos

Liste livros que não aparecem em itens_pedido (WHERE itens_pedido.livro_id IS NULL).

11 - Clientes sem pedidos

Liste clientes sem pedidos e mostre a cidade.nome.

12 - Desconto total por cupom

Para cada cupom ativo, calcule desconto_total = SUM(((percentual/100) * itens.quantidade * itens.preco_unitario) considerando apenas pedidos que usaram o cupom. Mostre cupom.codigo, ROUND(desconto_total, 2).

13 - TOP 5 categorias por faturamento

Some o faturamento bruto por categoria. Filtre com HAVING SUM(...) > 1000. Traga top 5 (ORDER BY total DESC LIMIT 5).

14 - Ticket médio por cliente

Calcule ticket_medio = SUM(valor_pedido)/COUNT(pedidos) onde valor_pedido é a soma de itens. Mostre cliente.nome, ROUND(ticket_medio, 2). HAVING COUNT(DISTINCT [pedidos.id](#)) >= 3.

15 - Distribuição por status

Conte pedidos por COALESCE(status, 'INDEFINIDO'). (Pode complementar com total geral via GROUP BY ROLLUP se quiser sofisticar.)

16 - Pagamentos por meio

Liste [pedido.id](#), pagamentos.meio, SUM(pagamentos.valor). Inclua pedidos

sem pagamento (COALESCE para 0). Agrupe de forma consistente.

17 - Estoque vs vendido

Mostre [livro.id](#), livro.titulo, estoque, vendido_total (soma de itens.quantidade), e CASE WHEN estoque - vendido_total < 0 THEN 'NEGATIVO' ELSE 'OK' END AS status_estoque.

18 - Hierarquia 2–3 níveis

Mostre cat_pai.nome, cat_filha.nome, cat_neta.nome (se existir). Use LEFT JOIN para não perder linhas sem neta.

19 - CROSS JOIN “controlado”

Gere cidades × vendedores e filtre no WHERE apenas combinações onde vendedor.cidade_id = [cidades.id](#) ou cidades.uf = 'SP'. Observe como o WHERE “estreita” o CROSS JOIN.

20- Clientes ativos no mês

Conte clientes com ≥ 1 pedido no mês corrente e clientes sem pedido no mês. Mostre ativos_mes e inativos_mes (pode ser uma única query com SUM(CASE ...)).