-- 1 Consulta de Funcionários

SELECT id\_vendedor AS id, nome , salario from VENDEDORES where inativo = FALSE order by nome;

-- 2 Funcionários com salário acima da média

SELECT id\_vendedor AS id, nome, salario FROM VENDEDORES WHERE salario > (SELECT AVG(salario) FROM VENDEDORES) ORDER BY salario DESC;

-- 3 Resumo por Cliente

-- Consulta otimizada

SELECT

c.ID\_CLIENTE AS id,

c.RAZAO\_SOCIAL AS razao\_social,

COALESCE(TRUNC(SUM(p.VALOR\_TOTAL)), 0) AS total

FROM

CLIENTES c

LEFT JOIN

PEDIDO p ON c.ID\_CLIENTE = p.ID\_CLIENTE

GROUP BY

c.ID\_CLIENTE, c.RAZAO\_SOCIAL

ORDER BY

total DESC; -- Ordenar por total de forma decrescente

-- 4

SELECT

id\_pedido AS id,

valor\_total AS valor,

data\_emissao AS data,

CASE

WHEN data\_cancelamento IS NOT NULL THEN 'CANCELADO'

WHEN data\_faturamento IS NOT NULL THEN 'FATURADO'

WHEN data\_faturamento IS NULL AND data\_cancelamento IS NULL THEN 'PENDENTE'

END AS situacao

FROM

PEDIDO;

-- 5

SELECT

ip.ID\_PRODUTO AS id\_produto,

SUM(ip.QUANTIDADE) AS quantidade\_vendida,

SUM(ip.QUANTIDADE \* ip.PRECO\_PRATICADO) AS total\_vendido,

COUNT(DISTINCT p.ID\_CLIENTE) AS clientes,

COUNT(DISTINCT ip.ID\_PEDIDO) AS pedidos

FROM

ITENS\_PEDIDO ip

JOIN

PEDIDO p ON ip.ID\_PEDIDO = p.ID\_PEDIDO

WHERE

p.DATA\_CANCELAMENTO IS NULL -- Exclui pedidos cancelados, se necessário

GROUP BY

ip.ID\_PRODUTO

ORDER BY

quantidade\_vendida DESC, total\_vendido DESC

LIMIT 1;