

Exercício de SQL

Tabelas:

Salesperson (ID, Name, Age, Salary)

Customer (ID, Name, City, IndustryType)

Orders (ID, order_date, customer_id, salesperson_id, Amount)

a) Retornar nomes de Salesperson que não tenham nenhum pedido do cliente "Samsonic".

-- Consulta:

```
SELECT s.Name
FROM Salesperson s
WHERE NOT EXISTS (
    SELECT 1
    FROM Orders o
    JOIN Customer c ON o.customer_id = c.ID
    WHERE o.salesperson_id = s.ID
    AND c.Name = 'Samsonic'
);
```

-- Explicação:

-- Usamos NOT EXISTS para filtrar vendedores que não possuem registros
-- em Orders associados ao cliente "Samsonic".

b) Atualizar nomes de Salesperson com 2 ou mais pedidos, adicionando '*' ao final.

-- Comando de atualização:

```
UPDATE Salesperson s
SET s.Name = s.Name || '*'
WHERE (
    SELECT COUNT(*)
    FROM Orders o
    WHERE o.salesperson_id = s.ID
) >= 2;
```

-- Explicação:

-- A subquery conta quantos pedidos cada vendedor tem. Se >=2,
-- concatenamos '*' ao campo Name.

c) Excluir todos os Salesperson que tenham feito pedidos para a cidade "Jackson".

-- Comando de exclusão:

```
DELETE FROM Salesperson
WHERE ID IN (
    SELECT o.salesperson_id
    FROM Orders o
    JOIN Customer c ON o.customer_id = c.ID
    WHERE c.City = 'Jackson'
);
```

-- Explicação:
-- A subconsulta retorna IDs de vendedores com pedidos para "Jackson".
-- Usamos IN para deletar esses registros na Salesperson.

d) Exibir total vendido por cada Salesperson. Se não tiver vendido nada, exibir zero.

-- Consulta de agregação:
SELECT s.Name,
 COALESCE(SUM(o.Amount), 0) AS TotalSold
FROM Salesperson s
LEFT JOIN Orders o ON s.ID = o.salesperson_id
GROUP BY s.Name
ORDER BY s.Name;

-- Explicação:
-- LEFT JOIN garante inclusão de todos os vendedores.
-- COALESCE converte NULL (nenhum pedido) em zero.