# **Exercícios de Fixação**

# **(8) WINDOW FUNCTIONS**

**1)** A cláusula OVER() é opcional na sintaxe das Window Functions para especificar as configurações de uma janela.

(V) Verdadeiro

(F) Falso

**2)** Podemos ter três tipos de Window Functions: Ranking, Agrregate e Offset.

(V) Verdadeiro

(F) Falso

**3)** Não podemos utilizar o PARTITION BY dentro de uma cláusula OVER nas Window Functions.

(V) Verdadeiro

(F) Falso

**4)** Explique o que as Window Functions fazem no comando abaixo (OBS: executar o comando):

SELECT

Id\_Venda, Id\_Cliente, Vl\_Venda,

ROW\_NUMBER() OVER(ORDER BY Vl\_Venda) AS [rownum],

RANK() OVER(ORDER BY Vl\_Venda) AS [rank],

DENSE\_RANK() OVER(ORDER BY Vl\_Venda) AS [dense\_rank],

NTILE(5) OVER(ORDER BY Vl\_Venda) AS [ntile]

FROM Vendas

ORDER BY Vl\_Venda

**5)** O NTILE irá dividir o resultado em grupos com 5 registros em cada um no máximo.

SELECT

Id\_Venda, Id\_Cliente, Vl\_Venda,

NTILE(5) OVER(ORDER BY Vl\_Venda) AS [ntile]

FROM Vendas

ORDER BY Vl\_Venda

(V) Verdadeiro

(F) Falso

**6)** O comando abaixo retorna em cada linha o valor das vendas individuais por cliente, o valor acumulado e o valor acumulado total.

SELECT

Id\_Venda, Id\_Cliente,

Vl\_Venda,

SUM(Vl\_Venda) OVER(PARTITION BY Id\_Cliente ORDER BY Id\_Venda) AS [soma\_acumulada],

SUM(Vl\_Venda) OVER(

PARTITION BY Id\_Cliente

ORDER BY Id\_Venda

ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING

) AS [soma\_total]

FROM Vendas

ORDER BY Id\_Cliente, Id\_Venda

(V) Verdadeiro

(F) Falso

**7)** A função “LAG” é utilizada para retornar a próxima venda, enquanto a função “LEAD” retorna a venda anterior.

SELECT

Id\_Venda, Id\_Cliente, Vl\_Venda,

LAG(Vl\_Venda) OVER(PARTITION BY Id\_Cliente ORDER BY Vl\_Venda) AS [lag],

LEAD(Vl\_Venda) OVER(PARTITION BY Id\_Cliente ORDER BY Vl\_Venda) AS [lead]

FROM Vendas

ORDER BY Id\_Cliente, Vl\_Venda

(V) Verdadeiro

(F) Falso