



Aula #7: Operadores de Lógica de Intervalo

☰ Módulo	SQL Básico
# Aula	13
🕒 Created	@May 13, 2022 5:29 PM
☑ Reviewed	☑
📎 Material PDF	

Aula #7: Operadores de Lógica de Intervalo

Objetivo da Aula:

- ☐ Entender como filtrar linhas a partir de intervalos
- ☐ Resumo
- ☐ Exercícios

Conteúdo:

▼ Operadores de Intervalo

Os operadores de intervalo selecionam as linhas da tabela dentro de um intervalo de valores. Os valores podem ser números, texto ou datas.

▼ 1. O operador **BETWEEN**

O operador **BETWEEN** permite a seleção das linhas dentro de um intervalo.

Quando aplicado a clausula WHERE, a consulta fica da seguinte forma: "Filtre todas as linhas cujo valor esteja entre A e B"

▼ Exemplo:

```
SELECT  
  COUNT( p.product_id )  
FROM products p  
WHERE p.product_category_name = 'audio'
```

```
AND p.product_photos_qty >= 3
AND p.product_photos_qty < 10
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_category_name = 'audio'
      AND p.product_photos_qty BETWEEN 3 AND 10
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_category_name = 'audio'
      AND p.product_photos_qty NOT BETWEEN 3 AND 10
```

**** O operador BETWEEN é uma alternativa para múltiplas condicionais AND.**

▼ 2. O operador IN

O operador **IN** permite o usuário especificar múltiplos valores para seleção.

Quando aplicado a clausula WHERE, a consulta fica da seguinte forma: **“Filtre todas as linhas cujo valor seja igual a X, Y ou Z”**

▼ Exemplo:

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_category_name = 'audio'
      OR p.product_category_name = 'perfumaria'
      OR p.product_category_name = 'artes'
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_category_name IN ( 'audio', 'perfumaria','artes' )
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_photos_qty IN ( 1, 2, 4, 7, 10 )
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_category_name NOT IN ( 'audio', 'perfumaria','artes' )
```

```
SELECT
  COUNT( p.product_id )
FROM products p
WHERE p.product_photos_qty NOT IN ( 1, 2, 4, 7, 10 )
```

▼ 3. O operador **LIKE**

O operador **LIKE** é usado para buscar um padrão no valor da coluna.

Quando aplicado a clausula WHERE, a consulta fica da seguinte forma: “Filtre todas as linhas caso você encontre um padrão no valor da coluna”

▼ Exemplo:

Filtra qualquer linha da **tabela “products”** onde o valor da **coluna “product_category_name”** comece com o **caracter “a”**, independente dos outros caracteres da palavra.

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
WHERE p.product_category_name LIKE 'a%'
GROUP BY p.product_category_name
```

Filtra qualquer linha da **tabela “products”** onde o valor da **coluna “product_category_name”** comece com os **caracteres “ar”**, independente dos outros caracteres da palavra.

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
WHERE p.product_category_name LIKE 'ar%'
GROUP BY p.product_category_name
```

Filtra qualquer linha da **tabela “products”** onde o valor da **coluna “product_category_name”** termine com os **caracteres “ria”**, independente dos outros caracteres da palavra

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
WHERE p.product_category_name LIKE '%ria'
GROUP BY p.product_category_name
```

Filtra qualquer linha da **tabela “products”** onde o valor da **coluna “product_category_name”** comece com **“a”**, tenha um caracter **“s” no meio** e **termine com “o”**, independente dos outros caracteres da palavra.

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
```

```
WHERE p.product_category_name LIKE 'a%s%'
GROUP BY p.product_category_name
```

▼ 4. O operador **HAVING**

O operador **HAVING** é usado para filtrar linhas a partir do resultado de uma função agregadora. Quando aplicado a clausula WHERE, a consulta fica da seguinte forma: “Filtre todas as linhas cujo valor resultante da soma da coluna X seja maior que Y”.

▼ Exemplo:

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
GROUP BY p.product_category_name
HAVING COUNT( p.product_id ) > 2000
```

```
SELECT
  p.product_category_name,
  COUNT( p.product_id ) AS produtos
FROM products p
WHERE p.product_category_name LIKE 'ar%'
GROUP BY p.product_category_name
HAVING COUNT( p.product_id ) > 50
```

```
SELECT
  p.product_category_name,
  SUM( p.product_weight_g ) AS peso
FROM products p
WHERE p.product_category_name LIKE 'ar%'
GROUP BY p.product_category_name
HAVING SUM( p.product_weight_g ) > 80000
```

▼ **Resumo:**

1. Os operadores de intervalo permitem selecionar linhas a partir de intervalos ou múltiplos valores definidos.
2. O operador LIKE permite a busca por padrões em texto.
3. O operador HAVING permite o filtro sobre o resultado de uma função agregadora.

▼ **Exercícios do PA Bond**

▼ **As perguntas do CEO:**

▼ **1. Quantos clientes únicos tiveram seu pedidos com status de “processing”, “shipped” e “delivered”, feitos entre os dias 01 e 31 de Outubro de 2016. Mostrar o resultado somente se o número total de clientes for acima de 5.**

```

SELECT
    order_status,
    COUNT( DISTINCT customer_id ) AS clientes_unicos
FROM orders o
WHERE order_purchase_timestamp BETWEEN '2016-10-01' AND '2016-10-31'
    AND order_status IN ( 'processing', 'shipped', 'delivered' )
GROUP BY order_status
HAVING COUNT( DISTINCT customer_id ) > 5

-- Resposta:
order_status  clientes_unicos
delivered     265
shipped       8

```

▼ 2. Mostre a quantidade total dos pedidos e o valor total do pagamento, para pagamentos entre 1 e 5 prestações ou um valor de pagamento acima de R\$ 5000.

```

SELECT
    payment_type,
    payment_installments,
    COUNT( op.order_id ) AS pedidos,
    SUM( op.payment_value ) AS valor_total_pagamento
FROM order_payments op
WHERE op.payment_installments NOT IN ( 2, 3, 4 )
    AND op.payment_value BETWEEN 0 AND 5000
GROUP BY payment_type, payment_installments
HAVING COUNT( op.order_id ) > 15000

-- Resposta:
payment_type  payment_installments  pedidos  valor_total_pagamento
boleto        1                      19780    2842355.9799999734
credit_card   1                      25455    2426781.3499999964

```

▼ 3. Quantos produtos estão cadastrados nas categorias: perfumaria, brinquedos, esporte lazer e cama mesa, que possuem entre 5 e 10 fotos, um peso que não está entre 1 e 5 g, um altura maior que 10 cm, uma largura maior que 20 cm. Mostra somente as linhas com mais de 10 produtos únicos.

```

SELECT
    product_category_name ,
    COUNT( DISTINCT product_id ) AS produtos_unicos
FROM products p
WHERE product_category_name IN ( 'perfumaria', 'brinquedos', 'esporte_lazer', 'cama_mesa_banho' )
    AND product_photos_qty BETWEEN 5 AND 10
    AND product_weight_g NOT BETWEEN 1 AND 5
    AND product_height_cm > 10
    AND product_width_cm > 20
GROUP BY product_category_name
HAVING COUNT( DISTINCT product_id ) > 10

-- Resposta:
product_category_name  produto
brinquedos             68
cama_mesa_banho        34
esporte_lazer          112

```

▼ 4. Refazer a consulta SQL abaixo, usando os operadores de intervalo.

```
SELECT
    order_status ,
    COUNT( order_id ) AS pedidos
FROM orders o
WHERE ( order_status = 'processing' OR order_status = 'canceled' )
    AND ( o.order_estimated_delivery_date > '2017-01-01' OR o.order_estimated_delivery_date < '2016-11-23' )
GROUP BY order_status

-- Resposta:
SELECT
    order_status ,
    COUNT( order_id ) AS pedidos
FROM orders o
WHERE order_status IN ( 'processing', 'canceled' )
    AND ( o.order_estimated_delivery_date > '2017-01-01' OR o.order_estimated_delivery_date < '2016-11-23' )
GROUP BY order_status
```

▼ 5. Qual a quantidade de cidades únicas dos vendedores no estado de São Paulo ou Rio de Janeiro com a latitude maior que -24.54 e longitude menor que -45.63?

```
SELECT
    g.geolocation_state,
    COUNT( DISTINCT g.geolocation_city ) AS cidades
FROM geolocation g
WHERE g.geolocation_state IN ( 'SP', 'RJ' )
    AND ( g.geolocation_lat > -24.54 AND g.geolocation_lng < -45.63 )
GROUP BY g.geolocation_state

-- Resposta:
RJ    9
SP   996
```

▼ 6. Quantos produtos estão cadastrados em qualquer categorias que comece com a letra “a” e termine com a letra “o” e que possuem mais de 5 fotos? Mostrar as linhas com mais de 10 produtos.

```
SELECT
    product_category_name ,
    COUNT( DISTINCT product_id ) AS produto
FROM products p
WHERE product_category_name LIKE 'a%'
    AND product_photos_qty > 5
GROUP BY product_category_name
HAVING COUNT( DISTINCT product_id ) > 10

-- Resposta:
product_category_name    produto
automotivo                102
```

▼ 7. Qual o número de clientes únicos, agrupados por estado e por cidades que comecem com a letra “m”, tem a letra “o” e terminem com a letra “a”? Mostrar os resultados somente para o número de clientes únicos maior que 10.

```

SELECT
    customer_state,
    c.customer_city,
    COUNT( DISTINCT c.customer_id ) AS cliente_unico
FROM customer c
WHERE c.customer_city LIKE 'm%o%a'
GROUP BY customer_state, customer_city
HAVING COUNT( DISTINCT c.customer_id ) > 10

-- Resposta:
customer_state customer_city      cliente_unico
MG              matias barbosa     13
SP              mairipora          75
SP              mococa             34
SP              mongagua           42
SP              monte azul paulista 19

```

▼ Na próxima aula ...

Aula #08: Operadores Condicionais