

Gabarito dos Exercícios – Aula 3

Curso de SQL para Análise de Dados

Exercícios e Gabarito

Ex. 1: Perfilamento de Dados: Liste a quantidade total de registros na tabela `customer` e a quantidade de valores distintos na coluna `last_name`.

```
SELECT COUNT(*) AS total_clientes ,  
       COUNT(DISTINCT last_name) AS sobrenomes_distintos  
FROM customer;
```

Ex. 2: Detecção de Nulos: Na tabela `customer`, verifique quantos clientes não possuem `email` informado.

```
SELECT COUNT(*) AS clientes_sem_email  
FROM customer  
WHERE email IS NULL;
```

Ex. 3: Distribuição de Frequências: Liste as 10 categorias (`category`) mais utilizadas em filmes.

```
SELECT c.name AS categoria ,  
       COUNT(*) AS total_filmes  
FROM film_category fc  
JOIN category c ON fc.category_id = c.category_id  
GROUP BY c.name  
ORDER BY total_filmes DESC  
LIMIT 10;
```

Ex. 4: Tratamento de Nulos com COALESCE: Na tabela `address`, substitua valores nulos de `postal_code`.

```
SELECT address_id ,  
       address ,  
       COALESCE(postal_code , 'N o Informado') AS  
       cep_tratado  
FROM address;
```

Ex. 5: Normalização de Categorias: Padronize os valores de `city` para maiúsculas.

```
SELECT city_id,
       UPPER(city) AS cidade_padronizada
FROM city;
```

Ex. 6: Conversão de Tipos: Na tabela `payment`, converta `payment_date` em apenas `DATE`.

```
SELECT payment_id,
       CAST(payment_date AS DATE) AS data_pagamento,
       amount
FROM payment;
```

Ex. 7: Binning de Valores Contínuos: Classifique os filmes da tabela `film` em faixas de duração.

```
SELECT title,
       length,
       CASE
         WHEN length < 60 THEN 'Curto (<60 min)'
         WHEN length BETWEEN 60 AND 120 THEN 'Médio (60-120 min)'
         ELSE 'Longo (>120 min)'
       END AS faixa_duracao
FROM film;
```

Ex. 8: Pivot de Vendas: Mostre o total de pagamentos por mês.

```
SELECT EXTRACT(MONTH FROM payment_date) AS mes,
       SUM(amount) AS total_vendas
FROM payment
GROUP BY EXTRACT(MONTH FROM payment_date)
ORDER BY mes;
```

Ex. 9: Unpivot: Transforme o resultado anterior em formato linha (mês/total).

```
-- O próprio resultado do Ex. 8 já fornece o formato "
  unpivotado":
SELECT TO_CHAR(payment_date, 'Month') AS mes,
       SUM(amount) AS total_vendas
FROM payment
GROUP BY TO_CHAR(payment_date, 'Month')
ORDER BY mes;
```

Ex. 10: CTE: Usando uma CTE, calcule o gasto médio por cliente e filtre quem gastou acima de 5.

```
WITH media_cliente AS (
  SELECT customer_id,
         AVG(amount) AS gasto_medio
  FROM payment)
```

```
        GROUP BY customer_id
    )
    SELECT c.customer_id,
           c.first_name,
           c.last_name,
           m.gasto_medio
    FROM media_cliente m
    JOIN customer c ON m.customer_id = c.customer_id
    WHERE m.gasto_medio > 5;
```