

# Teste Técnico - Analista de Dados

## Visão geral

Você trabalha para uma empresa que quer entender performance de vendas, produtos e clientes. O candidato receberá (ou gerará) três tabelas relacionais: customers, products e orders. O teste avalia: modelagem/SQL, capacidade analítica, qualidade de documentação, clareza do dashboard e storytelling.

None

```
CREATE TABLE customers (  
  customer_id INT PRIMARY KEY,  
  name VARCHAR(100),  
  email VARCHAR(100),  
  city VARCHAR(100),  
  created_at DATE  
);  
  
CREATE TABLE products (  
  product_id INT PRIMARY KEY,  
  name VARCHAR(100),  
  category VARCHAR(100),  
  price DECIMAL(10, 2)  
);  
  
CREATE TABLE orders (  
  order_id INT PRIMARY KEY,  
  customer_id INT,  
  product_id INT,  
  quantity INT,  
  order_date DATE,  
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id),  
  FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

## Dados sugeridos (opcional)

- Tamanho sugerido para testes de realismo: ~10k clientes, 200 produtos, 100k pedidos (últimos 24 meses).
- Inclua variação de preços por categoria e cidades com diferentes volumes (para testar agrupamentos).

## Parte 1 — SQL (entregáveis: scripts .sql)

Para cada item entregue:

- Indique o banco onde testou (Postgres / BigQuery / MySQL) e versões/peculiaridades.
- Forneça consulta SQL pronta para execução.
- Explique em 2–4 linhas a lógica e complexidade (por ex. índices que ajudariam).

### 1) Top clientes

- **Requisito:** Liste os 10 clientes que mais gastaram no total (considerar `price * quantity`).
- **Saída esperada (colunas):** `customer_id`, `name`, `total_spent`, `total_orders`, `avg_ticket`
- **Ordenação:** `total_spent` desc.
- **Observação:** trate nulos e múltiplas linhas por pedido.

### 2) Vendas por categoria

- **Requisito:** Para cada categoria, retorne receita total e quantidade total vendida.
- **Saída esperada:** `category`, `total_revenue`, `total_quantity`, `avg_price`
- **Ordenação:** receita desc.

### 3) Média de ticket por cidade

- **Requisito:** Ticket médio por pedido em cada cidade (`total_spent / total_orders` dentro da cidade).
- **Saída esperada:** `city`, `total_revenue`, `total_orders`, `avg_ticket`
- **Notas:** considerar apenas pedidos válidos; lidar com cidades vazias.

### 4) Evolução das vendas

- **Requisito:** Receita total por mês nos últimos 12 meses (último mês completo até a data mais recente nos dados).
- **Saída esperada:** `year_month` (YYYY-MM), `total_revenue`, `total_orders`

- **Formato:** ordenado por `year_month` asc.

## 5) Produto com maior crescimento (mês a mês)

- **Requisito:** No último semestre (últimos 6 meses), identifique o produto com maior crescimento em quantidade mês a mês.
- **Definição de crescimento:** soma das diferenças positivas entre meses consecutivos ou maior taxa média de crescimento (especifique a métrica escolhida).
- **Saída esperada:** `product_id, name, month_from, month_to, qty_from, qty_to, growth_absolute, growth_pct`
- **Observação:** descreva cálculo e limitações (ex.: sazonalidade, meses com 0 vendas).

## 6) Clientes inativos

- **Requisito:** Lista de clientes que não realizaram compras nos últimos 3 meses.
- **Saída esperada:** `customer_id, name, email, city, last_order_date`
- **Observação:** inclua clientes que nunca compraram se houver (marcar com `last_order_date IS NULL`).

# Parte 2 — Dashboard (entregáveis: link ou prints)

Ferramentas aceitáveis: Power BI, Looker Studio, Tableau, Metabase, etc. Preferir ferramenta que o candidato domine.

## Requisitos mínimos do dashboard

1. **KPI cards:** Total de receita, total de pedidos, ticket médio (receita/pedido), número de clientes (ativos no período).
2. **Gráfico de barras:** Receita por categoria (possibilidade de ordenar/filtrar).
3. **Gráfico de linha:** Receita por mês (últimos 12–24 meses) com opção de aplicar filtro de categoria ou cidade.
4. **Tabela interativa:** Clientes, com filtros por cidade e por período (sortable e com paginação).
5. **Gráfico de pizza / donut:** Distribuição de pedidos por cidade (ou top N cidades + “Outros”).
6. **Interatividade:** filtros por intervalo de datas, cidade e categoria; cross-filtering entre gráficos.
7. **Usabilidade & narrativa:** inclua um texto curto (1–3 parágrafos) explicando insights principais e recomendações.

## Entrega do dashboard

- Fornecer link público ou prints (alta resolução).

- Se a plataforma exigir acesso, conceder permissão de visualização (read-only).
- Incluir no README instruções para reproduzir (fonte dos CSVs, passos para carregar no Power BI/Looker).

## Entrega

- **Formato:** Link para repositório GitHub ou Google Drive com:
  - Scripts SQL.
  - Arquivos CSV (caso tenha criado dados próprios).
  - Documentação (README) com:
    - Ferramentas usadas e versões.
    - Como executar os scripts (comandos do psql, bq, etc).
    - Principais decisões, trade-offs e limitações dos dados/consultas.
    - Instruções para abrir o dashboard.
- **Prazo sugerido:** 3 a 5 dias.