# Relatório de Consultas SQL — Sistema de E-commerce

Este relatório apresenta uma análise detalhada das consultas SQL implementadas no sistema de e-commerce. Cada consulta foi desenvolvida para extrair informações específicas sobre clientes, produtos, pedidos e avaliações, visando proporcionar uma visão completa sobre o funcionamento e desempenho da plataforma.

---

## 1. Consulta de Clientes com seus respectivos Endereços

**SQL:**

```
SELECT c.id
        AS cliente_id, c.nome AS cliente_nome, c.cpf, c.email,
        c.telefone, c.dataNascimento, c.nacionalidade, c.genero, e.id
        AS endereco_id, e.cep, e.rua, e.complemento, e.logradouro,
        e.bairro, e.cidade, e.estado, e.numero
FROM cliente c
JOIN endereco e ON c.endereco_id = e.id;
```

**Descrição:** Esta consulta retorna uma lista de clientes com suas respectivas informações pessoais e endereços. Utiliza um `JOIN` entre as tabelas `cliente` e `endereco` para relacionar cada cliente ao seu endereço cadastrado. É útil para relatórios de usuários ou para verificar a abrangência geográfica dos clientes.

---

## 2. Consulta de Produtos com suas Avaliações

**SQL:**

```
SELECT a.nota, a.comentario, a.data, a.cliente_id,
        p.id AS produto_id, p.nome AS produto_nome,
        p.descricao, p.valor_atual, p.custo_atual,
        p.estoque
FROM avaliacao a
JOIN produto p ON a.produto_id = p.id;
```

**Descrição:** Esta consulta retorna uma lista de avaliações feitas pelos clientes, incluindo informações sobre o produto avaliado. Utiliza um `JOIN` entre as tabelas de `avaliacao` e `produto` para associar cada avaliação ao respectivo produto. Essa consulta é útil para

analisar o feedback dos clientes sobre os produtos, o que pode ajudar na gestão de estoque e na melhoria dos produtos com base nas avaliações.

---

## 3. Consulta de Carrinhos com Itens e Cliente

**SQL:**

```
SELECT ca.id AS carrinho_id,
       ca.data_criacao,
       cli.id AS cliente_id,
       cli.nome AS cliente_nome,
       cli.email,
       cli.telefone,
       ic.produto_id,
       ic.quantidade,
       ic.preco_unitario,
       p.nome AS produto_nome,
       p.descricao AS produto_descricao,
       p.valor_atual AS produto_valorAtual
FROM ecommerce.carrinho ca
JOIN ecommerce.cliente cli ON ca.cliente_id = cli.id
LEFT JOIN ecommerce.itensCarrinho ic ON ca.id = ic.carrinho_id
LEFT JOIN ecommerce.produto p ON ic.produto_id = p.id
ORDER BY ca.id;
```

**Descrição**: Esta consulta retorna os carrinhos de compras com seus respectivos itens e o cliente associado. Utiliza um `JOIN` entre as tabelas de `carrinho`, `cliente` e um `LEFT JOIN` entre `itensCarrinho` e `produto`. O resultado inclui informações sobre o cliente, o carrinho e os produtos contidos nele. A consulta é útil para analisar o comportamento de compra dos clientes e entender a composição dos carrinhos em termos de produtos e preços.

---

## 4. Quantidade de Pedidos por Cliente

**SQL:**

```
SELECT c.id AS cliente_id,
       c.nome,
       COUNT(v.id) AS total_pedidos
FROM ecommerce.cliente c
JOIN ecommerce.venda v
   ON c.id = v.cliente_id
GROUP BY c.id, c.nome
ORDER BY total_pedidos DESC;
```

**Descrição**: Esta consulta retorna a quantidade de pedidos realizados por cada cliente, agrupando os dados pela ID do cliente. Utiliza `JOIN` entre as tabelas `cliente` e `venda` agrupando por id do cliente e ordenando por total de pedidos. A consulta ajuda a entender o volume de compras de cada cliente, permitindo identificar clientes frequentes ou com maior engajamento. Também pode ser usada para ações de marketing direcionadas.

---

## 5. Média de Preço por Tipo de Pagamento

**SQL:**

```
SELECT v.tipoPagamento,
    AVG(v.total_pago) AS media_preco
FROM ecommerce.venda v
GROUP BY v.tipoPagamento
HAVING AVG(v.total_pago) > 50
ORDER BY media_preco DESC;
```

**Descrição**: Esta consulta calcula a média de preços dos produtos vendidos, agrupada por tipo de pagamento (como cartão de crédito, boleto, etc.). Utiliza a tabela `venda` agrupando por meio de pagamento. A consulta utiliza os dados de pedidos e forma de pagamento para calcular a média, oferecendo insights sobre as preferências de pagamento dos clientes e como isso pode afetar o preço médio dos produtos comprados.

---

## 6. Produtos Mais Vendidos

**SQL:**

```
SELECT iv.produto_id,
    p.nome,
    SUM(iv.quantidade) AS total_vendido
FROM ecommerce.itensvenda iv
LEFT JOIN ecommerce.produto p
  ON iv.produto_id = p.id
JOIN ecommerce.venda v
  ON iv.venda_id = v.id
WHERE iv.produto_id IS NOT NULL
GROUP BY iv.produto_id, p.nome
HAVING SUM(iv.quantidade) > 5
ORDER BY total_vendido DESC;
```

**Descrição**: Esta consulta retorna os produtos mais vendidos em um período específico. Utiliza `LEFT JOIN` entre as tabelas `itensvenda` e `produto` e um `JOIN` na tabela `venda`

agrupando por id do produto. A análise de quais produtos estão sendo mais vendidos pode fornecer dados úteis para ajustes de estoque e estratégias de marketing.

---

## 7 - Total de Pedidos no Último Mês

**SQL:**

```
SELECT COUNT(*) AS total_pedidos
FROM ecommerce.venda
WHERE data_hora >= DATE_SUB(CURDATE(), INTERVAL 1 MONTH);
```

**Descrição**: Esta consulta retorna o total de pedidos realizados no último mês. Ela utiliza a tabela de `vendas`, filtrando por data, para somar a quantidade total de pedidos feitos durante o mês anterior. Isso permite analisar o desempenho mensal das vendas e pode ser útil para relatórios de performance ou planejamento de estoque.

---

## 8 - Média de Vendas da Última Semana

**SQL:**

```
SELECT SUM(quantidade_pedidos) / 7.0 AS media_vendas_diarias
FROM (
    SELECT COUNT(*) AS quantidade_pedidos
    FROM ecommerce.venda v
    WHERE v.data_hora >= DATE_SUB(CURDATE(), INTERVAL 7 DAY)
    GROUP BY DATE(v.data_hora)
) AS vendas_diarias;
```

**Descrição**: Esta consulta calcula a média de vendas realizadas na última semana. Usando a tabela de `vendas` e filtrando e agrupando por data, a consulta fornece uma visão clara da performance semanal de vendas, o que pode ajudar a ajustar campanhas de marketing e estratégias de vendas.

---

## 9 - Clientes Ativos no Último Mês

**SQL:**

```
SELECT DISTINCT c.nome
FROM ecommerce.cliente c
JOIN ecommerce.venda v
   ON c.id = v.cliente_id
WHERE v.data_hora >= DATE_SUB(CURDATE(), INTERVAL 1
MONTH);
```

**Descrição**: Esta consulta retorna os clientes que realizaram pelo menos um pedido no último mês. Utilizando `join` entre a tabela de `cliente` e `venda`, ela filtra os clientes que realizaram compras, ajudando a identificar quais clientes estão ativos e comprometidos. Essas informações são valiosas para ações de retenção e fidelização de clientes.

---

## 10 - Cliente com Mais Pedidos

**SQL:**

```
SELECT c.nome, pv.total_pedidos
FROM ecommerce.cliente c
JOIN (
   SELECT cliente_id, COUNT(*) AS total_pedidos
   FROM ecommerce.venda
   GROUP BY cliente_id
   ORDER BY total_pedidos DESC
   LIMIT 1
) pv ON c.id = pv.cliente_id;
```

**Descrição**: Esta consulta identifica o cliente que realizou o maior número de pedidos. A consulta utiliza a tabela de `cliente` e um `JOIN` subconsulta na tabela `vendas` para contar a quantidade de pedidos feitos por cada cliente, permitindo identificar os clientes mais fieis ou mais engajados, o que pode ser útil para programas de fidelidade.

---

## 11 - Clientes com Pedidos Acima da Média

**SQL:**

```
SELECT nome
FROM ecommerce.cliente
WHERE id IN (
   SELECT cliente_id
   FROM ecommerce.venda
   WHERE total_pago > (SELECT AVG(total_pago) FROM
ecommerce.venda)
);
```

**Descrição**: Esta consulta retorna os clientes cujos pedidos estão acima da média dos valores de todos os pedidos. A média é calculada a partir da tabela de `cliente` junto a uma consulta aninhada `WHERE` por `id` na tabela `venda`, e a consulta filtra os clientes que gastaram mais do que esse valor médio. Essa informação pode ser útil para identificar clientes de alto valor para a empresa.

---

## 12 - Vendedores com Produtos Bem Avaliados

**SQL:**

```
SELECT v.nome, COUNT(p.id) AS quantidade_produtos
FROM vendedor v
JOIN produto p
   ON v.id = p.vendedor_id
WHERE p.id IN (
   SELECT produto_id
   FROM avaliacao
   GROUP BY produto_id
   HAVING AVG(nota) > 4
)
GROUP BY v.id;
```

**Descrição**: Esta consulta retorna os vendedores que possuem produtos que receberam as melhores avaliações. Utilizando um `JOIN` entre as tabelas de `vendedores` e `produto` junto a uma consulta aninhada `WHERE` por `produto_id` na tabela `avaliacao`, a consulta identifica quais produtos têm as avaliações mais altas. Isso pode ajudar a entender quais vendedores estão oferecendo produtos de alta qualidade e satisfação do cliente.

---

## 13 - Criação de clientes

**SQL:**

```
INSERT INTO cliente (nome, cpf, email, senha, telefone,
dataNascimento, nacionalidade, genero, endereco_id)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);
```

**Descrição**: Esta consulta é utilizada para inserir novos registros de clientes no sistema. Os valores “?” são passados como parâmetro dentro da query feita no sistema .Ela permite adicionar informações como nome, email, telefone e outros dados essenciais para o cadastro de um cliente. Essa consulta é importante para o processo de inclusão de novos clientes na base de dados.

---

## 14 - Atualização de clientes

**SQL:**

```
UPDATE cliente
SET nome = ?, cpf = ?, email = ?, senha = ?, telefone = ?,
dataNascimento = ?, nacionalidade = ?, genero = ?
WHERE id = ?;
```

**Descrição**: Esta consulta permite modificar os dados de clientes existentes no sistema. Os valores “?” são passados como parâmetro dentro da query feita no sistema. Ela pode ser utilizada para atualizar informações como nome, email, telefone e outros detalhes cadastrais de um cliente específico. Essa consulta é útil para manter os dados dos clientes atualizados e corrigir informações incorretas ou desatualizadas.

---

## 15 - Exclusão (física) de clientes

**Descrição**: Esta consulta é utilizada para excluir permanentemente um cliente do sistema. Ela remove todos os dados relacionados a um cliente, como nome, informações de contato e histórico de compras, de forma irreversível. É feita de forma onde somente apagar o endereço do cliente resulta na exclusão do cliente, uma vez que o endereço é vinculado unicamente a um cliente via chave estrangeira em modo cascata. A exclusão física é necessária quando o cliente decide não mais fazer parte da base de dados ou em casos de erro no cadastro.

---

# 16 - Leitura de clientes

**SQL:**

```
SELECT * FROM cliente WHERE id=?;
```

**Descrição**: Esta consulta retorna uma lista de todos os clientes cadastrados no sistema. Os valores “?” são passados como parâmetro dentro da query feita no sistema. Ela pode incluir informações básicas, como nome, ID e outros detalhes dos clientes. A consulta é útil para obter uma visão geral de todos os clientes registrados, sendo uma base para relatórios e análises relacionadas à base de usuários.

---

# 17 - Criação de endereços

**SQL:**

```
INSERT INTO endereco (cep, rua, complemento, logradouro, bairro,
cidade, estado, numero)
VALUES (?, ?, ?, ?, ?, ?, ?, ?);
```

**Descrição**: Esta consulta é utilizada para inserir novos registros de endereços para os clientes. Os valores “?” são passados como parâmetro dentro da query feita no sistema. Ela permite adicionar informações de endereço, como rua, número, cidade e estado, associando essas informações ao cliente correspondente. É importante para o processo de cadastro de novos endereços ou atualização dos dados de localização dos clientes.

---

# 18 - Edição de endereços

**SQL:**

```
UPDATE endereco
SET cep = ?, rua = ?, complemento = ?, logradouro = ?, bairro = ?,
cidade = ?, estado = ?, numero = ?
WHERE id = ?;
```

**Descrição**: Esta consulta permite modificar os dados de endereços existentes na tabela de endereços. Os valores “?” são passados como parâmetro dentro da query feita no sistema.

Ela pode ser utilizada para atualizar informações de rua, número, cidade ou outros dados relacionados ao endereço de um cliente específico. Essa consulta é útil quando há mudanças nas informações de endereço de clientes já cadastrados no sistema.

---

## 19 - Exclusão (física) de endereços

**SQL:**

```
DELETE FROM endereco
WHERE id = ?;
```

**Descrição**: Esta consulta é utilizada para remover um ou mais endereços cadastrados no sistema. Os valores “?” são passados como parâmetro dentro da query feita no sistema. Ela permite excluir registros de endereços específicos associados a um cliente. A exclusão pode ser necessária quando um cliente deseja desativar um endereço antigo ou quando há erro nos dados informados.

---

## 20 - Leitura de endereços

**SQL:**

```
SELECT * FROM endereco WHERE id=?;
```

**Descrição**: Esta consulta retorna uma lista de todos os endereços cadastrados no sistema, associados a seus respectivos clientes. Os valores “?” são passados como parâmetro dentro da query feita no sistema. Ela pode ser usada para visualizar ou analisar os locais de residência ou de entrega dos clientes. É útil para verificar a distribuição geográfica dos clientes ou preparar relatórios sobre a localização deles.