Tema: Consultas e operações avançadas em banco de dados relacional.

Total de Pontos: 40

Objetivo: Aplicar os conhecimentos de SQL sobre um banco de dados já modelado e populado, com foco em operações práticas e analíticas.

Justificativa/Argumentação

Neste trabalho, exploramos os principais recursos de SQL sobre um banco de dados relacional que seu grupo criou com registros e relacionamentos, permitindo a realização de consultas complexas e simulação de funcionalidades reais de sistemas de informação. Este projeto visa desenvolver a habilidade de manipulação e análise de dados utilizando instruções SQL, cobrindo desde comandos básicos até estruturas mais complexas como JOINs, subconsultas.

Tabela de Avaliação por Critério

Segue o código com o que se pede:

```
-- criação do banco de dados

create database sistema_vendas;

use sistema_vendas;


create table cliente (
   id_cliente int primary key,
   nome varchar(100),
   email varchar(100)
);


create table produto (
   id_produto int primary key,
   nome varchar(100),
   preco decimal(10,2)
);


create table pedido (
```

```
    id_pedido int primary key,
    data date,
    id_cliente int,
    id_produto int,
    foreign key (id_cliente) references cliente(id_cliente),
    foreign key (id_produto) references produto(id_produto)
);

create table pagamento (
    id_pg int primary key,
    id_pedido int,
    data date,
    valor decimal(10,2),
    foreign key (id_pedido) references pedido(id_pedido)
);

insert into cliente values
(1, 'Ana Silva', 'ana@email.com'),
(2, 'Bruno Costa', 'bruno@email.com'),
(3, 'Carlos Lima', 'carlos@email.com'),
(4, 'Daniela Souza', 'daniela@email.com'),
(5, 'Eduardo Alves', 'eduardo@email.com'),
(6, 'Fernanda Dias', 'fernanda@email.com'),
(7, 'Gustavo Rocha', 'gustavo@email.com'),
(8, 'Helena Torres', 'helena@email.com'),
(9, 'Igor Mendes', 'igor@email.com'),
(10, 'Juliana Lopes', 'juliana@email.com');
```

```
insert into produto values
(1, 'Teclado', 120.00),
(2, 'Mouse', 60.00),
(3, 'Monitor', 950.00),
(4, 'Notebook', 2800.00),
(5, 'Webcam', 150.00),
(6, 'Cadeira Gamer', 1100.00),
(7, 'Pen Drive', 35.00),
(8, 'HD Externo', 400.00),
(9, 'Impressora', 700.00),
(10, 'Microfone', 220.00);

insert into pedido values
(1, '2024-01-01', 1, 4),
(2, '2024-01-02', 2, 2),
(3, '2024-01-03', 3, 5),
(4, '2024-01-04', 4, 3),
(5, '2024-01-05', 5, 1),
(6, '2024-01-06', 6, 6),
(7, '2024-01-07', 7, 7),
(8, '2024-01-08', 8, 9),
(9, '2024-01-09', 9, 8),
(10, '2024-01-10', 10, 10);

insert into pagamento values
(1, 1, '2024-01-01', 2800.00),
(2, 2, '2024-01-02', 60.00),
(3, 3, '2024-01-03', 150.00),
```

```
(4, 4, '2024-01-04', 950.00),
(5, 5, '2024-01-05', 120.00),
(6, 6, '2024-01-06', 1100.00),
(7, 7, '2024-01-07', 35.00),
(8, 8, '2024-01-08', 700.00),
(9, 9, '2024-01-09', 400.00),
(10, 10, '2024-01-10', 220.00);

-- Q1 2 consultas com SELECT e WHERE 2,0 pts
-- 1. Clientes com nome iniciando por 'A'
select * from cliente where nome like 'A%';
-- 2. Produtos com preço maior que 500
select * from produto where preco > 500;

-- Q2 2 consultas com GROUP BY e ORDER BY com funções de agregação
-- 1. Total de pedidos por cliente
select id_cliente, count(*) as total_pedidos from pedido group by id_cliente
order by total_pedidos desc;
-- 2. Soma de pagamentos por data
select data, sum(valor) as total_pago from pagamento group by data order by data;

-- Q3 2 consultas com operadores aritméticos (+, -, *, /)
-- 1. Preço com 10% de desconto aplicado
select nome, preco, preco * 0.9 as preco_com_desconto from produto;
-- 2. Valor pago com acréscimo de taxa de 5%
select id_pg, valor, valor + (valor * 0.05) as valor_com_taxa from pagamento;

-- Q4 3 consultas com operadores de comparação (=, !=, <, >, etc.)
```

```sql
-- 1. Pedidos feitos após o dia 2024-01-05
select * from pedido where data > '2024-01-05';
-- 2. Produtos com preço diferente de 60
select * from produto where preco != 60;
-- 3. Pagamentos com valor menor que 500
select * from pagamento where valor < 500;

-- Q5 3 consultas com operadores lógicos (AND, OR)
-- 1. Produtos com preço acima de 500 E nome contendo "a"
select * from produto where preco > 500 and nome like '%a%';
-- 2. Clientes com nome 'Ana Silva' OU 'Carlos Lima'
select * from cliente where nome = 'Ana Silva' or nome = 'Carlos Lima';
-- 3. Pedidos após 2024-01-05 E com cliente 6
select * from pedido where data > '2024-01-05' and id_cliente = 6;

-- Q6 2 consultas com operadores lógicos e negação (NOT)
-- 1. Produtos que NÃO custam mais de 500
select * from produto where not preco > 500;
-- 2. Clientes que NÃO têm email @email.com
select * from cliente where not email like '%@email.com';

-- Q7 3 consultas com operadores auxiliares (IS NULL, BETWEEN, LIKE, IN)
-- 1. Produtos com preço entre 100 e 500
select * from produto where preco between 100 and 500;
-- 2. Clientes com nomes contendo 'o'
select * from cliente where nome like '%o%';
-- 3. Pagamentos com valor em uma lista específica
select * from pagamento where valor in (60.00, 220.00, 400.00);
```

```
-- Q8 3 consultas com funções de agregação (SUM(), AVG(), etc.)
-- 1. Total pago no sistema
select sum(valor) as total_geral from pagamento;
-- 2. Média de valor dos produtos
select avg(preco) as media_preco from produto;
-- 3. Produto mais caro
select max(preco) as mais_caro from produto;

-- Q9 2 consultas com funções de datas (NOW(), DATE(), YEAR(), etc.)
-- 1. Pedidos feitos no mesmo ano atual do sistema
select * from pedido where year(data) = year(now());
-- 2. Pagamentos feitos antes de hoje
select * from pagamento where data < date(now());

-- Q10 3 subconsultas com agrupamento e união de dados

-- 1. Lista de clientes que já fizeram pedidos (usando subconsulta)
select nome from cliente where id_cliente in (select id_cliente from pedido);
-- 2. Produtos com valor acima da média (subconsulta com agregação)
select * from produto where preco > (select avg(preco) from produto);
-- 3. União de dois conjuntos: produtos baratos e caros (UNION)
select nome, preco from produto where preco < 100
union
select nome, preco from produto where preco > 1000;

-- Q11 3 consultas com JOIN e visualização de tabelas
-- 1. Clientes e seus pedidos
```

```
select c.nome, p.id_pedido, p.data
from cliente c
join pedido p on c.id_cliente = p.id_cliente;
-- 2. Pedidos com nome do produto
select p.id_pedido, pr.nome as produto
from pedido p
join produto pr on p.id_produto = pr.id_produto;
-- 3. Pedidos e seus pagamentos
select pe.id_pedido, pa.valor
from pedido pe
join pagamento pa on pa.id_pedido = pe.id_pedido;

-- Q12 4 consultas com tipos de JOIN: INNER, LEFT, RIGHT
-- 1. INNER JOIN: pedidos com pagamento
select * from pedido p
inner join pagamento pa on pa.id_pedido = p.id_pedido;
-- 2. LEFT JOIN: todos os pedidos, pagos ou não
select * from pedido p
left join pagamento pa on pa.id_pedido = p.id_pedido;
-- 3. RIGHT JOIN: todos os pagamentos, mesmo sem pedido correspondente
select * from pedido p
right join pagamento pa on pa.id_pedido = p.id_pedido;
-- 4. INNER JOIN cliente + pedido + produto
select c.nome, pr.nome as produto
from cliente c
inner join pedido p on c.id_cliente = p.id_cliente
inner join produto pr on p.id_produto = pr.id_produto;
```