

Projeto Final - Sistema de Gerenciamento de Pedidos de Venda

1. Modelagem do Banco de Dados

As tabelas a serem criadas para o sistema são:

- **clientes:** id_cliente (PK), nome, email, telefone.
- **produtos:** id_produto (PK), nome, preco, estoque.
- **pedidos:** id_pedido (PK), id_cliente (FK), data_pedido.
- **pedido_produto:** id_pedido (FK), id_produto (FK), quantidade.

Script SQL:

```
CREATE TABLE clientes (  
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100),  
  email VARCHAR(100),  
  telefone VARCHAR(20)  
);
```

```
CREATE TABLE produtos (  
  id_produto INT AUTO_INCREMENT PRIMARY KEY,  
  nome VARCHAR(100),  
  preco DECIMAL(10,2),  
  estoque INT  
);
```

```
CREATE TABLE pedidos (  
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,  
  id_cliente INT,  
  data_pedido DATE,  
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)  
);
```

```
CREATE TABLE pedido_produto (  
  id_pedido INT,  
  id_produto INT,  
  quantidade INT,  
  PRIMARY KEY (id_pedido, id_produto),  
  FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido),  
  FOREIGN KEY (id_produto) REFERENCES produtos(id_produto)  
);
```

2. Relacionamentos

As chaves primárias e estrangeiras foram definidas no script acima, garantindo a integridade referencial entre as tabelas.

3. Consultas com JOINS e Subconsultas

- **JOIN** - Listar pedidos com clientes e produtos:

```
SELECT
  p.id_pedido,
  c.nome AS cliente,
  pr.nome AS produto,
  pp.quantidade,
  p.data_pedido
FROM pedidos p
JOIN clientes c ON p.id_cliente = c.id_cliente
JOIN pedido_produto pp ON p.id_pedido = pp.id_pedido
JOIN produtos pr ON pp.id_produto = pr.id_produto;
```

- **Subconsulta** - Produtos mais vendidos:

```
SELECT nome, total_vendido FROM (
  SELECT
    pr.nome,
    SUM(pp.quantidade) AS total_vendido
  FROM pedido_produto pp
  JOIN produtos pr ON pp.id_produto = pr.id_produto
  GROUP BY pr.nome
) AS vendas
ORDER BY total_vendido DESC
LIMIT 5;
```

4. Procedimento Armazenado

Criação do procedimento para registrar pedido:

```
DELIMITER $$
```

```
CREATE PROCEDURE registrar_pedido (
```

```

    IN cliente_id INT,
    IN produto_id INT,
    IN qtd INT
)
BEGIN
    DECLARE id_pedido_novo INT;

    INSERT INTO pedidos (id_cliente, data_pedido)
    VALUES (cliente_id, CURDATE());

    SET id_pedido_novo = LAST_INSERT_ID();

    INSERT INTO pedido_produto (id_pedido, id_produto, quantidade)
    VALUES (id_pedido_novo, produto_id, qtd);

    UPDATE produtos
    SET estoque = estoque - qtd
    WHERE id_produto = produto_id;
END$$

DELIMITER ;

```

Execução:

```
CALL registrar_pedido(1, 2, 5);
```

5. Otimização de Consultas

- Criação de índices:

```

CREATE INDEX idx_cliente_id ON pedidos(id_cliente);
CREATE INDEX idx_produto_id ON pedido_produto(id_produto);

```

- Análise com EXPLAIN:

```
EXPLAIN SELECT * FROM pedidos WHERE id_cliente = 1;
```

- Boas práticas:
Evitar:

```
WHERE UPPER(nome) = 'JOÃO'
```

Utilizar:

WHERE nome = 'João'

Esse projeto integra conceitos de modelagem, consultas SQL, procedimentos e boas práticas de desempenho, simulando um sistema real de vendas.