

## PRÁTICA SQL

Insira neste documento os comandos SQL de cada questão.

1. Crie um novo banco de dados e dê um nome a ele.

```
CREATE DATABASE loja
```

2. Crie neste banco a tabela “cliente”, com os atributos idcliente, nome, sobrenome, cpf, sendo o idcliente uma chave primária.

```
CREATE TABLE cliente (idcliente INT NOT NULL, nome VARCHAR(40), sobrenome VARCHAR(40), cpf int NOT NULL, PRIMARY KEY (idcliente))
```

3. Crie a tabela “pagamento” neste banco, com os atributos idpgto, valor e data sendo o idpgto a chave primária e idcliente uma chave estrangeira da tabela pagamento.

```
CREATE TABLE pagamento ( idpgto INT NOT NULL, valor VARCHAR(40), data VARCHAR(40), idcliente INT NOT NULL, PRIMARY KEY(idpgto), FOREIGN KEY(idcliente) REFERENCES cliente(idcliente) )
```

4. Insira três clientes e para cada cliente, um a dois pagamentos.

```
INSERT INTO cliente VALUES (1, 'joao', 'silva', 12345678909), (2, 'maria', 'silva', 98765432198), (3, 'adinam', 'fonseca', 11122233344)
```

```
INSERT INTO pagamento VALUES (11, 5000, '2024-07-18', '1'), (22, 6000, '2024-08-01', '2'), (33, 10000, '2024-06-06', '3')
```

5. Altere a data de um pagamento.

```
UPDATE pagamento SET data = '2024-07-20' WHERE idcliente = 1
```

6. Delete um cliente. Relate o que houve e por quê.

```
DELETE FROM cliente  
WHERE idcliente = 2;
```

Ao tentar excluir um cliente, ocorreu um erro indicando que não foi possível realizar a operação de exclusão devido a uma restrição de chave estrangeira. Isso ocorreu porque existem registros na tabela pagamento que estão associados ao cliente que está sendo excluído. A integridade referencial está sendo violada, pois a tabela pagamento possui uma chave estrangeira que referencia a tabela cliente.

7. Apresente a soma de todos os pagamentos realizados.

```
SELECT SUM(valor) AS soma_pagamento FROM pagamento
```

8. Utilize o LIKE para listar os dados de alguns clientes.

```
SELECT * FROM cliente WHERE sobrenome LIKE '%silva'
```

9. Utilize o GROUP BY para uma consulta.

```
SELECT idcliente, COUNT(*) AS valor FROM pagamento GROUP BY idcliente
```

10. Elabore uma consulta listando o nome do cliente com os dados do pagamento.

```
SELECT c.nome AS NomeCliente, p.idpgto AS IdPagamento, p.valor AS ValorPagamento,  
p.data AS DataPagamento  
FROM cliente c, pagamento p  
WHERE c.idcliente = p.idcliente
```

11. Transcreva esta consulta usando JOIN.

```
SELECT c.nome AS NomeCliente, p.idpgto AS IdPagamento, p.valor AS ValorPagamento,  
p.data AS DataPagamento  
FROM cliente c  
JOIN pagamento p ON c.idcliente = p.idcliente
```

12. Elaborar uma nova consulta utilizando dados de duas tabelas.

```
SELECT c.nome, p.valor, p.data FROM cliente c JOIN pagamento p ON c.idcliente=p.idcliente
```