

Desafio 02

01 A empresa quer melhorar o cadastro dos clientes, adicionando rua, numero, complemento, bairro, cidade e estado. Como efetuar essa mudança na tabela?

```
ALTER TABLE enderecos
ADD COLUMN rua VARCHAR(100),
ADD COLUMN bairro VARCHAR(100),
ADD COLUMN numero INT,
ADD COLUMN complemento VARCHAR(20),
ADD COLUMN cidade VARCHAR(100),
ADD COLUMN estado VARCHAR(100)
```

02 A empresa quer uma totalização de clientes agrupado por estado e por tipo de conta. Qual a consulta para obter isso?

```
SELECT enderecos.estado, tipos_conta.identificacao, COUNT(DISTINCT clientes.cliente_id) FROM clientes
JOIN clientes_conta
    ON clientes.cliente_id = clientes_conta.cliente_id
JOIN contas
    ON clientes_conta.nro_conta = contas.nro_conta
JOIN tipos_conta
    ON contas.tipo_conta_id = tipos_conta.tipo_conta_id
JOIN cliente_enderecos
    ON clientes.cliente_id = cliente_enderecos.cliente_id
JOIN enderecos
    ON cliente_enderecos.endereco_id = enderecos.endereco_id
GROUP BY enderecos.estado, tipos_conta.identificacao
ORDER BY enderecos.estado ASC
```

03 O centro de regulação bancária decidiu que as contas bancárias de empresas não podem mais ter clientes dependentes. A empresa pediu para efetuarmos uma copia dos dados desses clientes e depois apagar esses dados. Como proceder?

```
SELECT * INTO clientes_backup FROM clientes
```

04 A empresa decidiu que, todo primeiro dia do mês, vai enviar um email para todos os cliente premium que façam aniversário no mês corrente, desejando feliz aniversário. Para isso você precisa mandar esses dados para o marketing todos os meses. Como resolver este problema?

dica: a função `date_part('month', now())`; retorna um número inteiro relativo ao mês do momento da execução da query.

```
-- Adicionar data de aniversário na tabela de clientes
ALTER TABLE clientes
ADD COLUMN data_aniversario TIMESTAMP

-- Selecionar clientes que fazem aniversário no mês
CREATE VIEW "Aniversariantes do Mês" AS
SELECT * FROM clientes
WHERE EXTRACT(MONTH FROM data_aniversario) = DATE_PART('month', NOW())
```

05 Quanto dinheiro a empresa tem sob custódia?

```
SELECT SUM(valor) FROM transacoes
```

06 Qual é a média de volume financeiro transacionado por tipo de transação nos últimos dois meses? (volume transacionado é a soma de todos os valores transacionados, independente de entrada ou saída. Se houve um depósito de 100 e uma retirada de 20, o volume transacionado é 120)

```
SELECT tipos_transacao.identificacao, AVG(ABS(valor)) FROM transacoes
JOIN tipos_transacao
    ON transacoes.tipo_transacao_id = tipos_transacao.tipo_transacao_id
WHERE transacoes.data > DATE_TRUNC('day', NOW() - INTERVAL '2 month')
GROUP BY tipos_transacao.identificacao
```

07 O presidente da empresa decidiu cobrir todas as contas negativas, deixando os saldos dos clientes com saldo negativo em 0. Quanto dinheiro é necessário para fazer isso, e como efetuar esse processo?

```
-- Adicionar coluna saldo na tabela de contas
```

```
ALTER TABLE contas
ADD COLUMN saldo float
```

```
-- Calcular valor de saldo menor que zero
```

```
SELECT SUM(saldo) FROM contas
WHERE saldo < 0
```

```
-- Replace
```

```
UPDATE contas
SET saldo = 0
WHERE saldo < 0
```

```
-- Ou
```

```
SELECT clientes.cliente_id, clientes.nome, clientes.cpf, SUM(transacoes.valor)
FROM clientes
JOIN clientes_conta
    ON clientes.cliente_id = clientes_conta.cliente_id
LEFT JOIN transacoes
    ON clientes_conta.nro_conta = transacoes.nro_conta
GROUP BY clientes.cliente_id
HAVING SUM(transacoes.valor) < 0
```

08 As dez contas pessoais com mais saldo em conta serão atualizadas para conta premium. Como fazer essa alteração?

```
-- Selecionando contas com maior saldo
```

```
SELECT saldo, tipo_conta_id FROM contas
ORDER BY saldo DESC
LIMIT 10
```

```
-- Atualizando tipo de conta
```

```
UPDATE contas
SET tipo_conta_id = 3
WHERE saldo IN (
    SELECT saldo
    FROM contas
    ORDER BY saldo DESC
    LIMIT 10
)
```

```
-- Ou
```

```
SELECT clientes.cliente_id, clientes.nome, clientes.cpf, SUM(transacoes.valor)
FROM clientes
JOIN clientes_conta
    ON clientes.cliente_id = clientes_conta.cliente_id
LEFT JOIN transacoes
    ON clientes_conta.nro_conta = transacoes.nro_conta
GROUP BY clientes.cliente_id
ORDER BY SUM(transacoes.valor) desc
```

LIMIT 10

09 As contas corporativas passam, a partir de amanhã, a ter um gerente que cuida dos interesses da empresa junto ao banco. Um gerente pode cuidar de mais de uma conta, mas uma conta só pode ter um gerente por vez. Os gerentes de conta podem ser mudados quando necessário. Quais alterações são necessárias no banco de dados?

```
CREATE TABLE gerentes (  
  gerente_id INT NOT NULL UNIQUE,  
  nome VARCHAR(100) NOT NULL,  
  cpf CHAR(11) NOT NULL UNIQUE,  
  created_at TIMESTAMP NOT NULL DEFAULT NOW(),  
  modified_at TIMESTAMP,  
  deleted_at TIMESTAMP,  
  PRIMARY KEY (gerente_id)  
)
```

```
CREATE TABLE gerente_enderecos (  
  gerente_id INT NOT NULL,  
  endereco_id INT NOT NULL,  
  principal BOOL DEFAULT FALSE,  
  created_at TIMESTAMP NOT NULL DEFAULT NOW(),  
  modified_at TIMESTAMP,  
  deleted_at TIMESTAMP,  
  PRIMARY KEY (gerente_id, endereco_id),  
  FOREIGN KEY (gerente_id) REFERENCES gerentes(gerente_id),  
  FOREIGN KEY (endereco_id) REFERENCES enderecos(endereco_id)  
)
```

```
CREATE TABLE gerente_conta (  
  nro_agencia INT NOT NULL,  
  nro_conta INT NOT NULL,  
  gerente_id INT NOT NULL,  
  cliente_id INT NOT NULL,  
  created_at TIMESTAMP NOT NULL DEFAULT NOW(),  
  modified_at TIMESTAMP,  
  deleted_at TIMESTAMP,  
  PRIMARY KEY (nro_agencia, nro_conta, cliente_id, gerente_id),  
  FOREIGN KEY (nro_conta, nro_agencia) REFERENCES contas (nro_conta, nro_agencia),  
  FOREIGN KEY (gerente_id) REFERENCES gerentes (gerente_id),  
  FOREIGN KEY (cliente_id) REFERENCES clientes (cliente_id)  
)
```

10 Entendemos que um cliente está pra fechar a conta quando tem pouco dinheiro e quase não movimenta a conta. Por isso, queremos nos precaver em relação a isso. Como eu posso gerar um relatório diário de clientes com pouco saldo e que está há mais tempo sem efetuar transações?

```
-- Selecionando contas com menor saldo e transações mais antigas  
CREATE VIEW "Relatório Clientes" AS  
SELECT DISTINCT clientes.cliente_id, clientes.nome, contas.saldo, transacoes.data FROM clientes  
JOIN clientes_conta  
  ON clientes.cliente_id = clientes_conta.cliente_id  
JOIN contas  
  ON clientes_conta.nro_conta = contas.nro_conta  
JOIN transacoes  
  ON contas.nro_conta = transacoes.nro_conta  
ORDER BY contas.saldo ASC, transacoes.data ASC
```