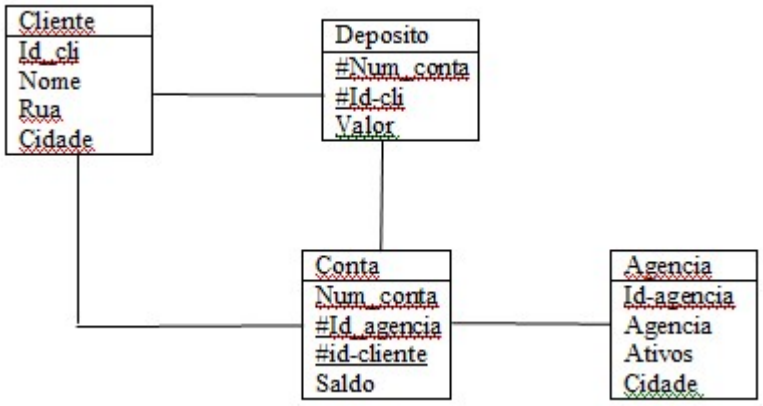
1 Um determinado banco da cidade de Miracema do Norte decidiu automatizar o seu sistema de controle bancário. Para isso o banco contratou a empresa EngSoft 3B para o desenvolvimento e suporte do software. A empresa de análise do sistema decidiu pelo seguinte esquema:



O banco de dados criado, já com algumas informações cadastradas, deve retornar algumas consultas, consideradas de suma importância para o banco.

Você, como integrante da equipe da empresa EngSoft 3B, deverá criar as consultas capazes de satisfazer o cliente nos critérios apresentados. Atenção, você poderá usar o banco de dados para fazer o teste, mas você deverá **registrar as consultas** neste relatório para avaliação do cliente, juntamente com o **resultado que a consulta** retorna.

1.1 Mostrar a soma dos depósitos feitos na conta 445544 e o saldo atual desta conta

```
MariaDB [controle_bancario]> CREATE VIEW soma_deposito01 AS
    -> SELECT d.num_conta, SUM(d.valor) AS soma_depositos, c.saldo + SUM(d.valor) AS saldo_Total
    -> FROM deposito d
    -> LEFT JOIN conta c ON c.num_conta = d.num_conta
    -> WHERE c.num_conta = '445544'
    -> GROUP BY d.num_conta, c.saldo;
Query OK, 0 rows affected (0.008 sec)

MariaDB [controle_bancario]> SELECT num_conta, soma_depositos, saldo_Total
    -> FROM soma_deposito01;
+-----------+----------------+-------------+
| num_conta | soma_depositos | saldo_Total |
+-----------+----------------+-------------+
|    445544 |         600.00 |      800.00 |
+-----------+----------------+-------------+
1 row in set (0.004 sec)
```

1.2 Buscar a soma dos saldos dos clientes da agência 8 e quantas contas existem nessa agencia.

```
MariaDB [controle_bancario]> SELECT SUM(c.saldo) AS Soma_Total_Saldos, COUNT(*) AS Totais_Contas
    -> FROM conta c
    -> INNER JOIN agencia a ON a.id_agencia = c.id_agencia
    -> WHERE a.id_agencia = 8;
+-------------------+---------------+
| Soma_Total_Saldos | Totais_Contas |
+-------------------+---------------+
|            250.00 |             1 |
+-------------------+---------------+
1 row in set (0.012 sec)
```

1.3 Mostre o nome dos clientes que tem conta nas agências de Joinville com saldo menor que R$ 300,00 e que não moram em Joinville

```
MariaDB [controle_bancario]> SELECT cli.nome
    -> FROM cliente cli
    -> INNER JOIN conta c ON c.id_cliente = cli.id_cli
    -> INNER JOIN agencia a ON a.id_agencia = c.id_agencia
    -> WHERE a.cidade = 'Joinville' AND c.saldo < 300.00 AND cli.cidade <> 'Joinville';
+--------+
| nome   |
+--------+
| Marcos |
+--------+
1 row in set (0.001 sec)
```

2 Quando apresentado ao diretor do banco a base de dados atual, o mesmo sentiu a necessidade de uma automatização no processo de atualização do saldo da conta bancária toda vez que houver uma remoção de depósito na referida conta.
Assim, você ficou responsável pela criação deste gatilho lançado toda vez que esta operação for executada na base de dados.

```
MariaDB [controle_bancario]> DELIMITER &
MariaDB [controle_bancario]>
MariaDB [controle_bancario]> CREATE TRIGGER Atualizar_Saldo_Conta
    -> AFTER DELETE ON deposito
    -> FOR EACH ROW
    -> BEGIN
    ->     DECLARE valor_removido DECIMAL(20,2);
    ->     SET valor_removido = OLD.valor;
    ->     UPDATE conta SET saldo = saldo - valor_removido WHERE num_conta = OLD.num_conta;
    -> END &
Query OK, 0 rows affected (0.032 sec)

MariaDB [controle_bancario]>
MariaDB [controle_bancario]> DELIMITER ;
MariaDB [controle_bancario]>
```

3 A direção do banco pretende criar um programa de fidelidade. Para isso ele sugere a inserção de pontos associada ao cliente cada vez que o mesmo fizer uma operação de depósito na sua conta bancária. Como você pode solucionar isso? Mostra as alterações e os comandos necessários para atender este pedido.

```
MariaDB [controle_bancario]>
MariaDB [controle_bancario]> DELIMITER //
MariaDB [controle_bancario]>
MariaDB [controle_bancario]> CREATE FUNCTION atualizar_pontos_cliente(num_conta INT) RETURNS INT
    -> BEGIN
    ->     DECLARE pontos_atual INT;
    ->     SELECT SUM(valor) INTO pontos_atual FROM deposito WHERE num_conta = num_conta;
    ->     UPDATE cliente SET pontos = pontos_atual WHERE id_cli = (SELECT id_cliente FROM conta WHERE num_conta = num_conta);
    ->     RETURN pontos_atual;
    -> END //
Query OK, 0 rows affected (0.012 sec)

MariaDB [controle_bancario]>
MariaDB [controle_bancario]> DELIMITER ;
MariaDB [controle_bancario]>
MariaDB [controle_bancario]>
MariaDB [controle_bancario]> DELIMITER //
MariaDB [controle_bancario]>
MariaDB [controle_bancario]> CREATE TRIGGER atualizar_pontos_deposito
    -> AFTER INSERT ON deposito
    -> FOR EACH ROW
    -> BEGIN
    ->     CALL atualizar_pontos_cliente(NEW.num_conta);
    -> END //
Query OK, 0 rows affected (0.008 sec)

MariaDB [controle_bancario]>
MariaDB [controle_bancario]> DELIMITER ;
MariaDB [controle_bancario]>
```

4 Antes do término do projeto, o diretor do banco solicitou a criação de uma visão que, ao ser consultada, retornasse o número e o nome da agência, o nome do cliente, e o seu saldo principal ordenado pelo nome da agência.

```
MariaDB [controle_bancario]> CREATE VIEW saldoAgencia AS
    -> SELECT a.id_agencia, a.agencia, cli.nome AS nome_cliente, c.saldo
    -> FROM agencia a
    -> INNER JOIN conta c ON c.id_agencia = a.id_agencia
    -> INNER JOIN cliente cli ON cli.id_cli = c.id_cliente
    -> ORDER BY a.agencia;
Query OK, 0 rows affected (0.005 sec)

MariaDB [controle_bancario]> SELECT id_agencia, agencia, nome_cliente, saldo
    -> FROM saldoAgencia;
+------------+-----------+--------------+--------+
| id_agencia | agencia   | nome_cliente | saldo  |
+------------+-----------+--------------+--------+
|          1 | Agencia 1 | Edson        | 200.00 |
|          1 | Agencia 1 | Marcos       | 100.00 |
|          2 | Agencia 2 | Maria        | 100.00 |
|          3 | Agencia 3 | Carlos       | 300.00 |
|          4 | Agencia 4 | Pedro        | 500.00 |
|          5 | Agencia 5 | Clara        | 100.00 |
|          6 | Agencia 6 | Livia        |  50.00 |
|          7 | Agencia 7 | Renata       | 150.00 |
|          8 | Agencia 8 | Andre        | 250.00 |
+------------+-----------+--------------+--------+
9 rows in set (0.006 sec)
```