SISTEMA NULLBANK

Professor: Fernando Rodrigues

João Marcos Rocha Souza 521459

Priscila Áquila Araujo Morais 499464

UNIVERSIDADE FEDERAL DO CEARÁ

Sumário

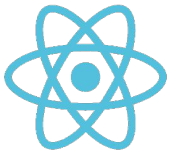
1 Tecnologias utilizadas na aplicação

2 Modelagem do Banco

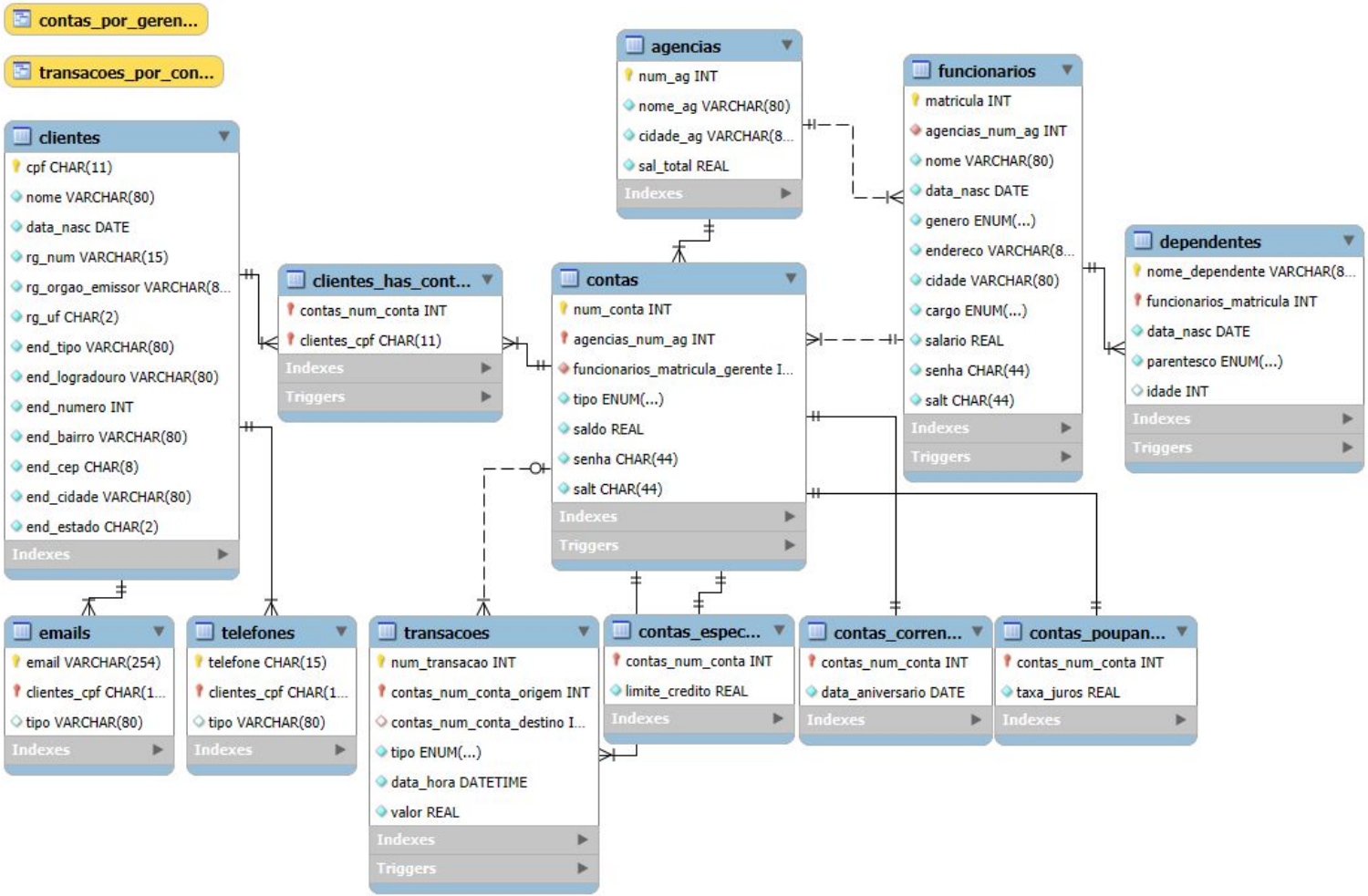
3 Triggers

Tecnologias

Tecnologias utilizadas na aplicação



Modelagem do Banco


contas_por_geren...
transacoes_por_con...
clientes
cpf CHAR(11)
nome VARCHAR(80)
data_nasc DATE
rg_num VARCHAR(15)
rg_orgao_emissor VARCHAR(8...
rg_uf CHAR(2)
end_tipo VARCHAR(80)
end_logradouro VARCHAR(80)
end_numero INT
end_bairro VARCHAR(80)
end_cep CHAR(8)
end_cidade VARCHAR(80)
end_estado CHAR(2)
Indexes
clientes_has_cont...
contas_num_conta INT
clientes_cpf CHAR(11)
Indexes
Triggers
agencias
num_ag INT
nome_ag VARCHAR(80)
cidade_ag VARCHAR(8...
sal_total REAL
Indexes
contas
num_conta INT
agencias_num_ag INT
funcionarios_matricula_gerente I...
tipo ENUM(...)
saldo REAL
senha CHAR(44)
salt CHAR(44)
Indexes
Triggers
funcionarios
matricula INT
agencias_num_ag INT
nome VARCHAR(80)
data_nasc DATE
genero ENUM(...)
endereco VARCHAR(8...
cidade VARCHAR(80)
cargo ENUM(...)
salario REAL
senha CHAR(44)
salt CHAR(44)
Indexes
Triggers
dependentes
nome_dependente VARCHAR(8...
funcionarios_matricula INT
data_nasc DATE
parentesco ENUM(...)
idade INT
Indexes
Triggers
emails
email VARCHAR(254)
clientes_cpf CHAR(1...
tipo VARCHAR(80)
Indexes
telefones
telefone CHAR(15)
clientes_cpf CHAR(1...
tipo VARCHAR(80)
Indexes
transacoes
num_transacao INT
contas_num_conta_origem INT
contas_num_conta_destino I...
tipo ENUM(...)
data_hora DATETIME
valor REAL
Indexes
Triggers
contas_espec...
contas_num_conta INT
limite_credito REAL
Indexes
contas_corren...
contas_num_conta INT
data_aniversario DATE
Indexes
contas_poupan...
contas_num_conta INT
taxa_juros REAL
Indexes

Triggers

Triggers

(Pt.1)

```
1  -- Limita em 2 a quantidade de clientes que podem possuir a mesma conta
2  -- Limita em 1 a quantidade de contas que um cliente pode ter em uma mesma agência
3
4  CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`clientes_has_contas_BEFORE_INSERT` BEFORE INSERT ON `clientes_has_contas` FOR EACH ROW
5  BEGIN
6      DECLARE qtd_clientes_conta INT;
7      DECLARE cliente_has_conta_in_agencia INT;
8
9      -- Conta quantos clientes já possuem a conta que está sendo associada
10     SELECT COUNT(*) INTO qtd_clientes_conta
11     FROM clientes_has_contas
12     WHERE contas_num_conta = NEW.contas_num_conta;
13
14     IF qtd_clientes_conta >= 2 THEN
15         SIGNAL SQLSTATE '45000'
16         SET MESSAGE_TEXT = 'Uma conta não pode ter mais de dois clientes associados a ela.';
17     END IF;
```

Triggers

(Pt.2)

```
19      -- Conta quantas contas o cliente tem na agência da conta que está sendo associada a ele
20      SELECT COUNT(*) INTO cliente_has_conta_in_agencia
21      FROM clientes_has_contas, contas
22      WHERE
23          contas_num_conta = num_conta AND
24          clientes_cpf = NEW.clientes_cpf AND
25          agencias_num_ag = (
26              SELECT agencias_num_ag
27              FROM contas
28              WHERE num_conta = NEW.contas_num_conta
29          );
30
31      IF cliente_has_conta_in_agencia > 0 THEN
32          SIGNAL SQLSTATE '45000'
33          SET MESSAGE_TEXT = 'Um cliente só pode ter uma conta por agência.';
34      END IF;
35  END
```

Triggers

```sql
-- Verifica se o funcionário associado a conta a ser inserida possui o cargo de 'gerente'
-- Verifica se o gerente associado a conta pertence a mesma agência a qual a conta está sendo associada

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`contas_BEFORE_INSERT` BEFORE INSERT ON `contas` FOR EACH ROW
BEGIN
    DECLARE funcionario_cargo VARCHAR(10);
    DECLARE funcionario_agencia INT;

    SELECT cargo, agencias_num_ag INTO funcionario_cargo, funcionario_agencia
    FROM funcionarios
    WHERE matricula = NEW.funcionarios_matricula_gerente;

    IF funcionario_cargo <> 'gerente' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Somente um funcionário com cargo de "gerente" pode ser associado a gerente de uma conta.';
    END IF;

    IF funcionario_agencia <> NEW.agencias_num_ag THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'O gerente da conta deve pertencer a mesma agência da conta.';
    END IF;
END
```

Triggers

```sql
-- Verifica se o funcionário associado a conta a ser atualizada possui o cargo de 'gerente'
-- Verifica se o gerente associado a conta pertence a mesma agência a qual a conta está sendo associada

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`contas_BEFORE_UPDATE` BEFORE UPDATE ON `contas` FOR EACH ROW
BEGIN
    DECLARE funcionario_cargo VARCHAR(10);
    DECLARE funcionario_agencia INT;

    -- Só executa se o gerente da conta tiver sido alterado
    IF OLD.funcionarios_matricula_gerente <> NEW.funcionarios_matricula_gerente THEN
        SELECT cargo, agencias_num_ag INTO funcionario_cargo, funcionario_agencia
        FROM funcionarios
        WHERE matricula = NEW.funcionarios_matricula_gerente;

        IF funcionario_cargo <> 'gerente' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Somente um funcionário com cargo de "gerente" pode ser associado a gerente de uma conta.';
        END IF;

        IF funcionario_agencia <> NEW.agencias_num_ag THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'O gerente da conta deve pertencer a mesma agência da conta.';
        END IF;
    END IF;
END
```

Triggers

```sql
-- Confere o piso salarial do funcionário inserido

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`funcionarios_BEFORE_INSERT` BEFORE INSERT ON `funcionarios` FOR EACH ROW
BEGIN
    IF NEW.salario < 2286.00 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Um funcionário não pode ter o salário menor que 2.286,00.';
    END IF;
END
```

Triggers

```sql
-- Atualiza o sal_total da agência em que o funcionário foi inserido

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`funcionarios_AFTER_INSERT` AFTER INSERT ON `funcionarios` FOR EACH ROW
BEGIN
    UPDATE agencias
    SET sal_total = (
        SELECT SUM(salario)
        FROM funcionarios
        WHERE agencias_num_ag = NEW.agencias_num_ag
    )
    WHERE num_ag = NEW.agencias_num_ag;
END
```

Triggers

```sql
-- Confere o piso salarial do funcionário atualizado

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`funcionarios_BEFORE_UPDATE` BEFORE UPDATE ON `funcionarios` FOR EACH ROW
BEGIN
    IF NEW.salario < 2286.00 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Um funcionário não pode ter o salário menor que 2.286,00.';
    END IF;
END
```

Triggers (Pt.1)

```sql
-- Atualiza o sal_total da agência do funcionário atualizado

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`funcionarios_AFTER_UPDATE` AFTER UPDATE ON `funcionarios` FOR EACH ROW
BEGIN
    -- TRIGGER Executa somente se o salário OU a agência do funcionário tiver sido alterado(a)
    IF OLD.salario <> NEW.salario OR OLD.agencias_num_ag <> NEW.agencias_num_ag THEN
        -- Atualiza o sal_total da antiga agência do funcionário
        UPDATE agencias
        SET sal_total = (
            SELECT SUM(salario)
            FROM funcionarios
            WHERE agencias_num_ag = OLD.agencias_num_ag
        )
        WHERE num_ag = OLD.agencias_num_ag;
```

Triggers (Pt.2)

```
16          -- Se a agência tiver sido alterada, atualiza também o sal_total da nova agência
17          IF OLD.agencias_num_ag <> NEW.agencias_num_ag THEN
18              UPDATE agencias
19              SET sal_total = (
20                  SELECT SUM(salario)
21                  FROM funcionarios
22                  WHERE agencias_num_ag = NEW.agencias_num_ag
23              )
24              WHERE num_ag = NEW.agencias_num_ag;
25          END IF;
26      END IF;
27  END
```

Triggers

```sql
-- Atualiza o sal_total da agência do funcionário deletado

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`funcionarios_AFTER_DELETE` AFTER DELETE ON `funcionarios` FOR EACH ROW
BEGIN
    UPDATE agencias
    SET sal_total = (
        SELECT SUM(salario)
        FROM funcionarios
        WHERE agencias_num_ag = OLD.agencias_num_ag
    )
    WHERE num_ag = OLD.agencias_num_ag;
END
```

Triggers

```sql
-- Limita em 5 a quantidade de dependentes que um funcionário pode ter

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`dependentes_BEFORE_INSERT` BEFORE INSERT ON `dependentes` FOR EACH ROW
BEGIN
    DECLARE num_dependentes INT;

    -- Conta a quantidade de dependentes do funcionário que está sendo associado ao dependente inserido
    SELECT COUNT(nome_dependente) INTO num_dependentes
    FROM dependentes
    WHERE funcionarios_matricula = NEW.funcionarios_matricula;

    IF num_dependentes >= 5 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Um funcionário não pode ter mais que 5 dependentes.';
    END IF;

    SET NEW.idade = TIMESTAMPDIFF(YEAR, NEW.data_nasc, CURDATE());

END
```

Triggers

```
1  CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`dependentes_BEFORE_UPDATE` BEFORE UPDATE ON `dependentes` FOR EACH ROW
2  BEGIN
3      SET NEW.idade = TIMESTAMPDIFF(YEAR, NEW.data_nasc, CURDATE());
4  END
```

Triggers - Transações Before (Pt.1)

```sql
-- Verifica se a conta associada a transação possui saldo para faze-la
-- Verifica se transferência e PIX possuem conta de destino

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`transacoes_BEFORE_INSERT` BEFORE INSERT ON `transacoes` FOR EACH ROW
BEGIN
    DECLARE saldo_atual REAL;
    DECLARE limite_credito_especial REAL;

    -- Somente movimentações de saída precisam de saldo em conta
    IF NEW.tipo IN('saque', 'pagamento', 'transferência', 'PIX') THEN
        SELECT saldo INTO saldo_atual
        FROM contas
        WHERE num_conta = NEW.contas_num_conta_origem;

        -- Obtem o limite de crédito se a conta for especial
        SELECT limite_credito INTO limite_credito_especial
        FROM contas_especial
        WHERE contas_num_conta = NEW.contas_num_conta_origem;

        -- Caso a conta não seja especial, o limite de crédito é 0
        IF limite_credito_especial IS NULL THEN
            SET limite_credito_especial = 0;
        END IF;
```

Triggers - Transações Before (Pt.2)

```
25          -- Saldo em conta + limite de crédito especial (se houver) devem ser maior ou igual ao valor da transação
26          IF (saldo_atual + limite_credito_especial) < NEW.valor THEN
27              SIGNAL SQLSTATE '45000'
28              SET MESSAGE_TEXT = 'A conta não possui saldo e/ou crédito para realizar esta transação.';
29          END IF;
30      END IF;
31
32      -- Verifica se transferência e PIX possuem conta de destino
33      IF NEW.tipo IN('transferência', 'PIX') AND NEW.contas_num_conta_destino IS NULL THEN
34          SIGNAL SQLSTATE '45000'
35          SET MESSAGE_TEXT = 'Transferência e PIX devem possuir uma conta de destino.';
36      END IF;
37  END
```

Triggers - Transações After (Pt.1)

```sql
-- Atualiza o saldo da conta e o limite de crédito (se for o caso) ao inserir uma transação

CREATE DEFINER = CURRENT_USER TRIGGER `Equipe521459`.`transacoes_AFTER_INSERT` AFTER INSERT ON `transacoes` FOR EACH ROW
BEGIN
    DECLARE current_saldo REAL;
    DECLARE value_to_sub_on_saldo REAL;
    DECLARE value_to_sub_on_limite_credito REAL;

    -- Transações que debitam saldo da conta ou do limite crédito
    IF NEW.tipo IN('saque', 'pagamento', 'transferência', 'PIX') THEN
        -- Obtem o saldo atual
        SELECT saldo INTO current_saldo
        FROM contas
        WHERE num_conta = NEW.contas_num_conta_origem;

        -- Verifica qual o valor a ser debitado do saldo e do limite de crédito se for necessário
        IF current_saldo - NEW.valor < 0 THEN
            SET value_to_sub_on_saldo = current_saldo;
            SET value_to_sub_on_limite_credito = NEW.valor - value_to_sub_on_saldo;
        ELSE
            SET value_to_sub_on_saldo = NEW.valor;
        END IF;
```

Triggers - Transações After (Pt.2)

```
24          -- Se houver débito de saldo, atualiza o saldo da conta
25          IF value_to_sub_on_saldo > 0 THEN
26              UPDATE contas
27              SET saldo = saldo - value_to_sub_on_saldo
28              WHERE num_conta = NEW.contas_num_conta_origem;
29          END IF;
30
31          -- Se houver dábito de crédito, atualiza o limite de crédito
32          IF value_to_sub_on_limite_credito > 0 THEN
33              UPDATE contas_especial
34              SET limite_credito = limite_credito - value_to_sub_on_limite_credito
35              WHERE contas_num_conta = NEW.contas_num_conta_origem;
36          END IF;
37
38          -- Transferência e PIX subtraem da conta origem e somam ao saldo da conta destino
39          IF NEW.tipo IN ('transferência','PIX') THEN
40              UPDATE contas
41              SET saldo = saldo + NEW.valor
42              WHERE num_conta = NEW.contas_num_conta_destino;
43          END IF;
44      END IF;
```

Triggers - Transações After (Pt.3)

```
46      -- Deposito e estorno somam ao saldo da conta
47      IF NEW.tipo IN ('deposito','estorno') THEN
48          UPDATE contas
49          SET saldo = saldo + NEW.valor
50          WHERE num_conta = NEW.contas_num_conta_origem;
51      END IF;
52  END
```

Obrigado pela atenção!