

Trabalho Final de Banco de Dados 2024.2  
Professor: Fernando Rodrigues

# NullBank

Ana Lara Carvalho Mesquita – 498866  
Maria Camilly do Vale Carneiro - 509751

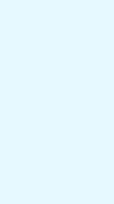


UFC

# Sistema bancário NullBank

O banco NullBank é organização bancária que necessita de um sistema bancário para coletar, gerenciar e armazenar as informações que são necessárias para o funcionamento do banco, tais informações como clientes, contas, agências, funcionário e outras que são valiosas para a organização.

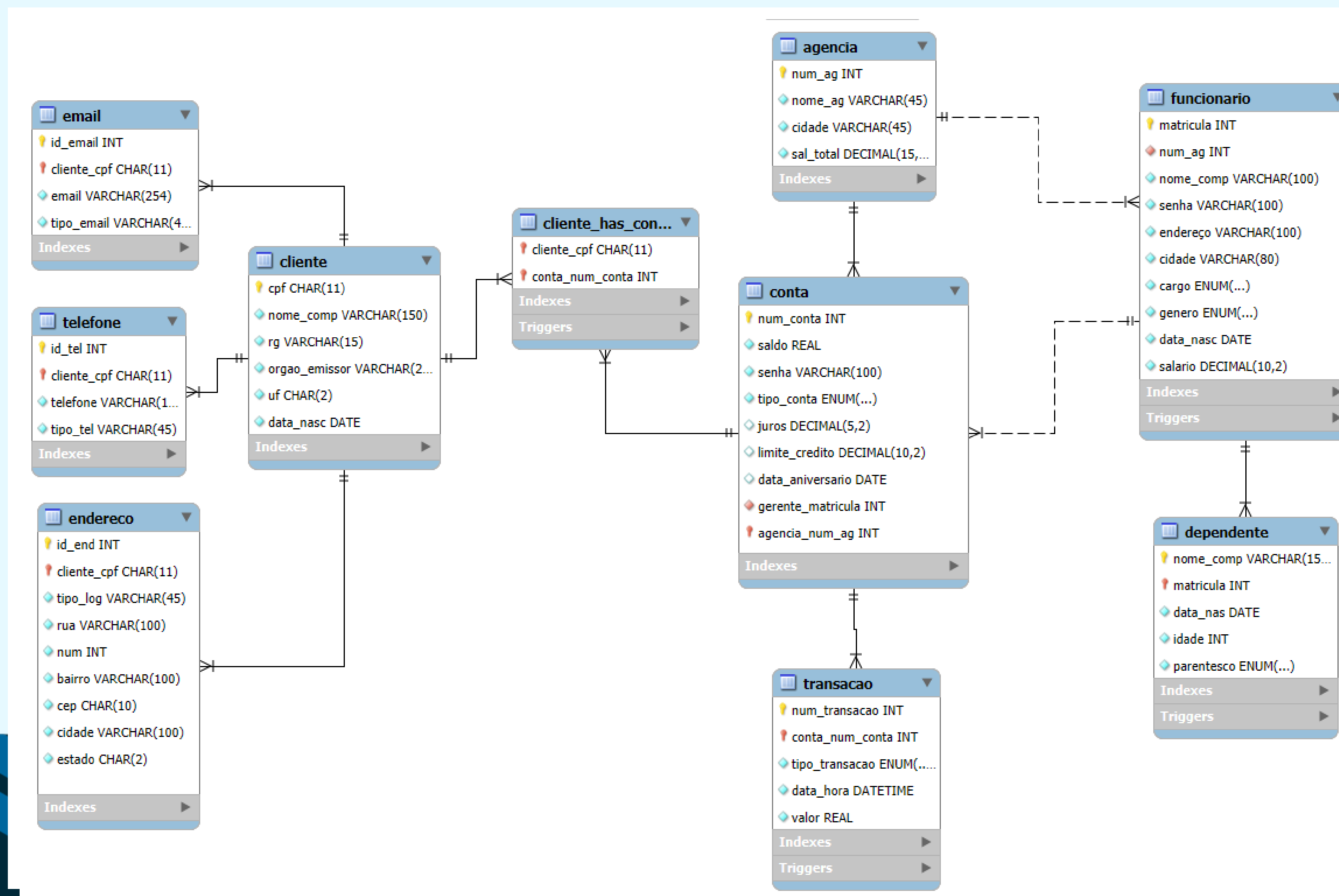




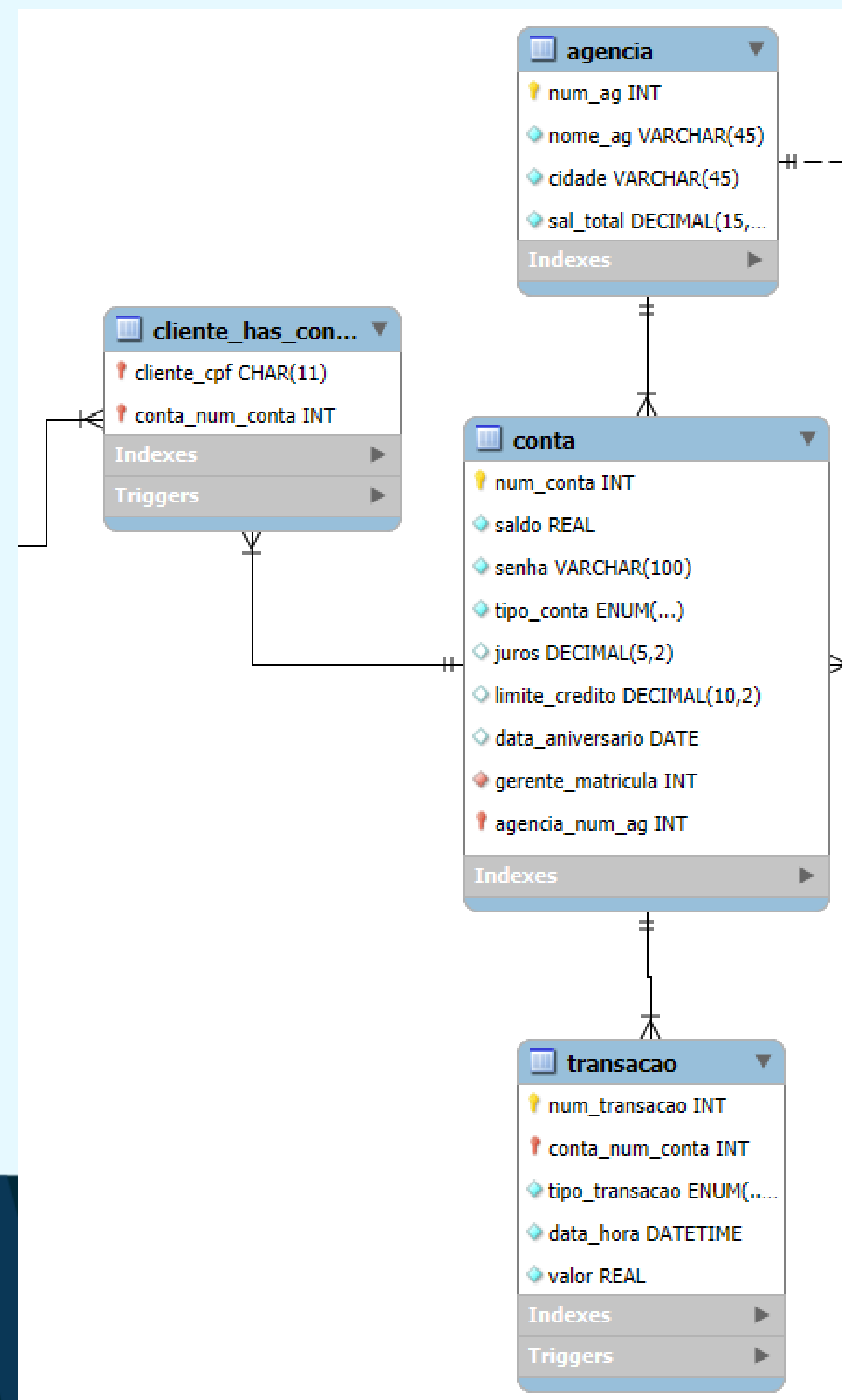
# Tecnologias

- MySQL
- MySQL Workbench

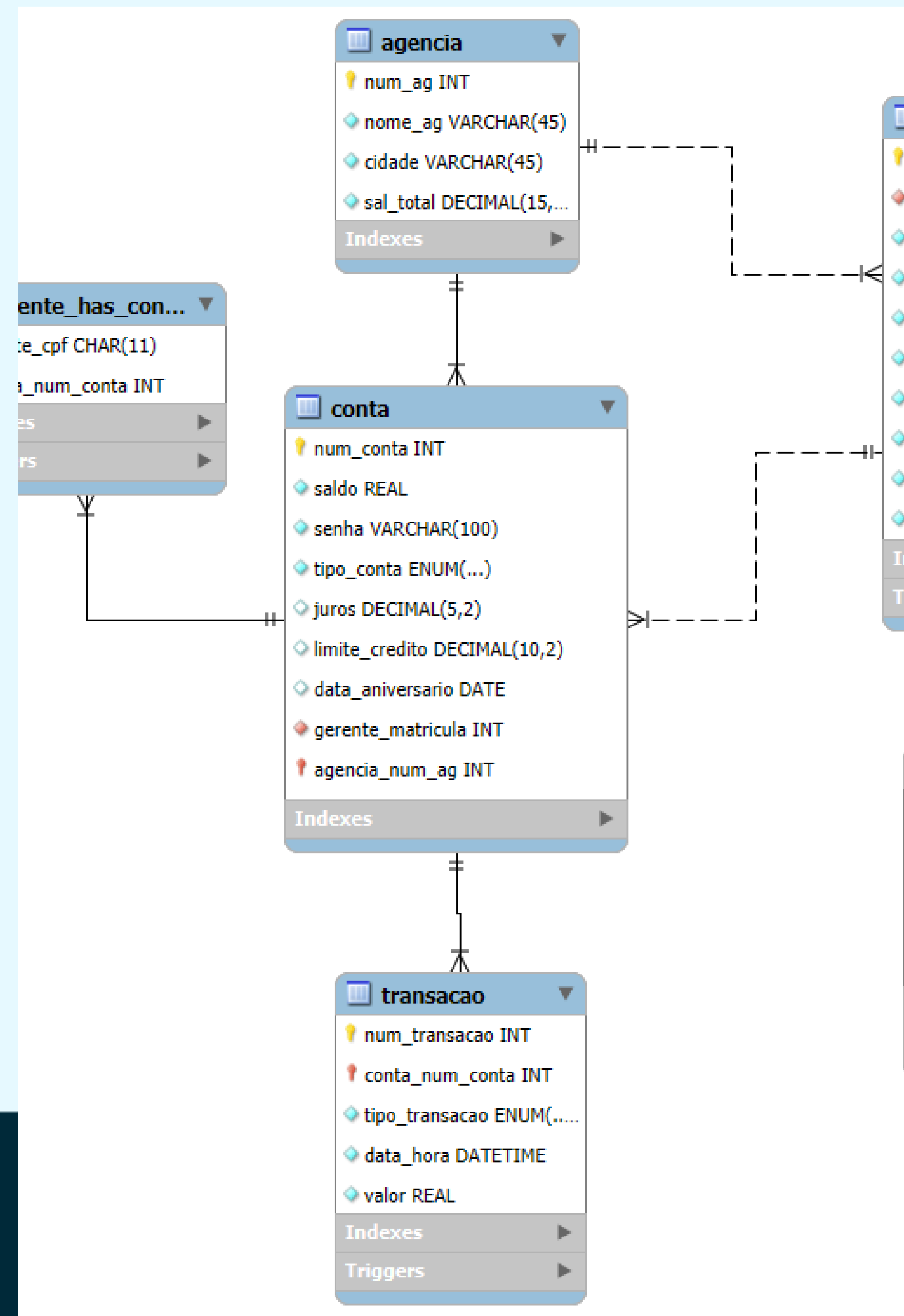
# Modelagem



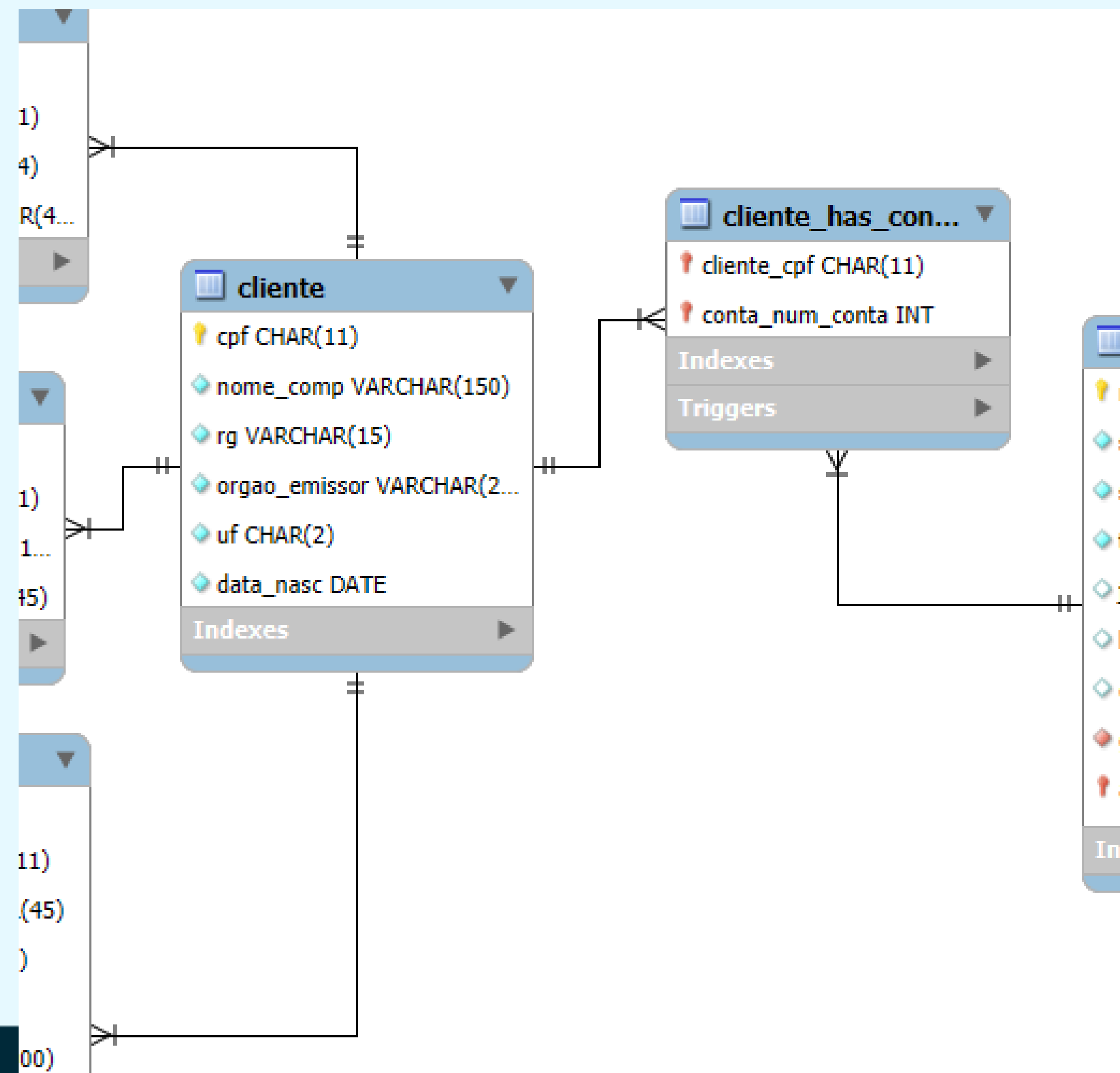
# Funcionário e Dependente



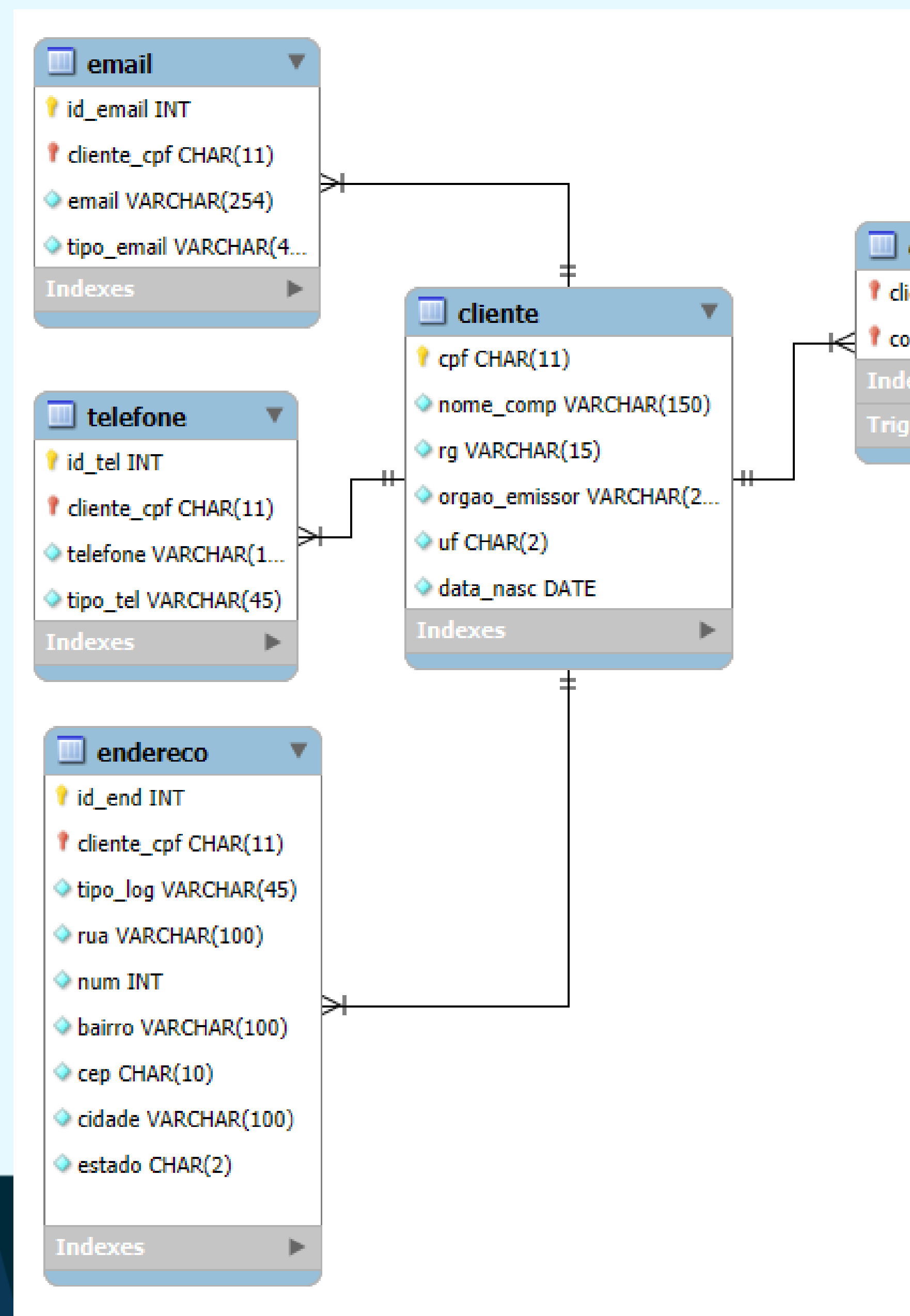
# Agência, Conta e Transação



# Cliente e Cliente\_has\_Conta



# Cliente, Email, Telefone e Endereço





# Triggers – Funcionário

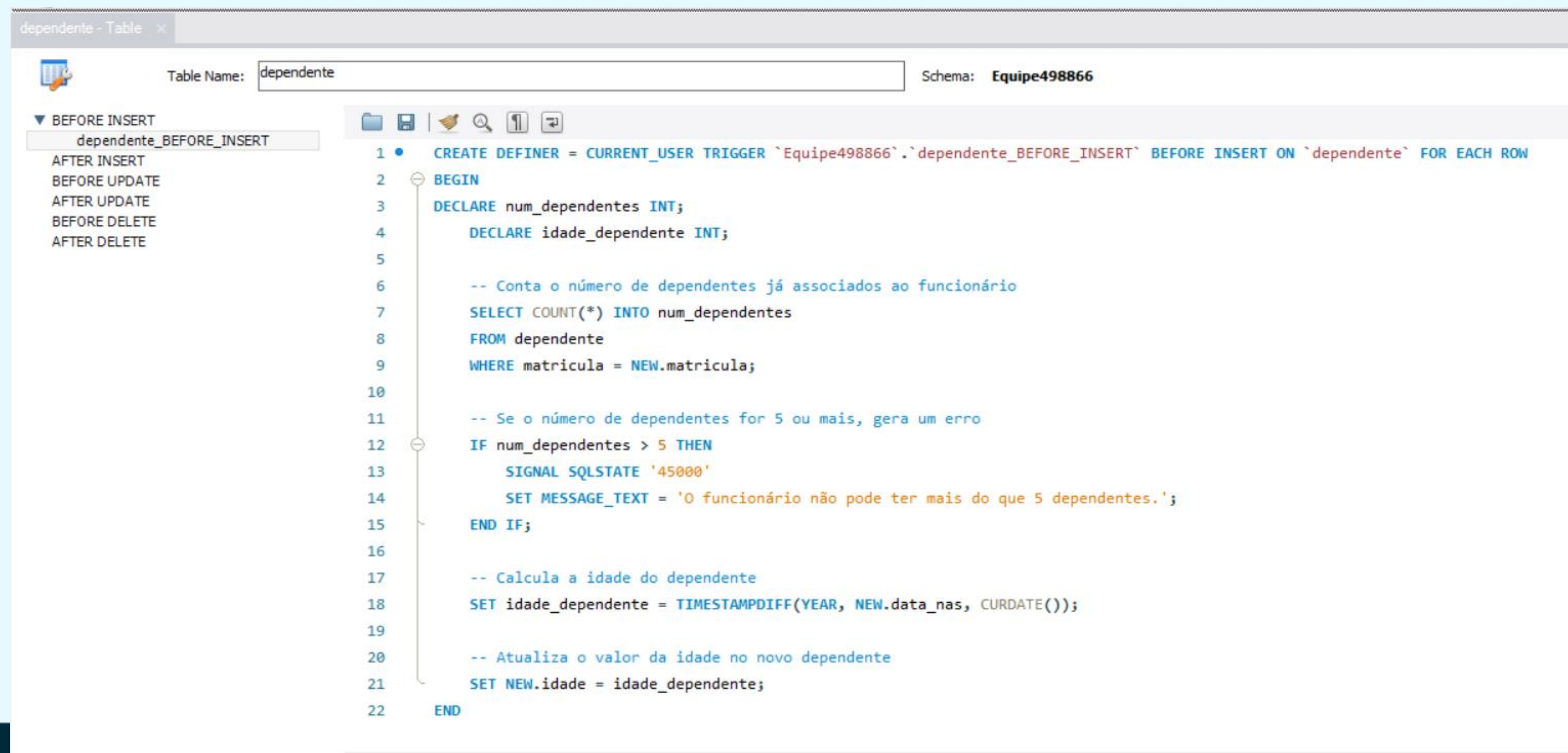
funcionario - Table x

Table Name:  Schema: **Equipe498866**

▼ BEFORE INSERT  
funcionario\_BEFORE\_INSERT  
▼ AFTER INSERT  
funcionario\_AFTER\_INSERT  
▼ BEFORE UPDATE  
funcionario\_BEFORE\_UPDATE  
▼ AFTER UPDATE  
funcionario\_AFTER\_UPDATE  
▼ BEFORE DELETE  
funcionario\_BEFORE\_DELETE  
AFTER DELETE

```
1 • CREATE DEFINER = CURRENT_USER TRIGGER `Equipe498866`.`funcionario_BEFORE_INSERT` BEFORE INSERT ON `funcionario`  
2 BEGIN  
3 IF NEW.salario < 2286.00 THEN  
4     SIGNAL SQLSTATE '45000'  
5     SET MESSAGE_TEXT = 'O salário não pode ser menor que R$2.286,00.';  
6 END IF;  
7 END
```

# Triggers – Dependente

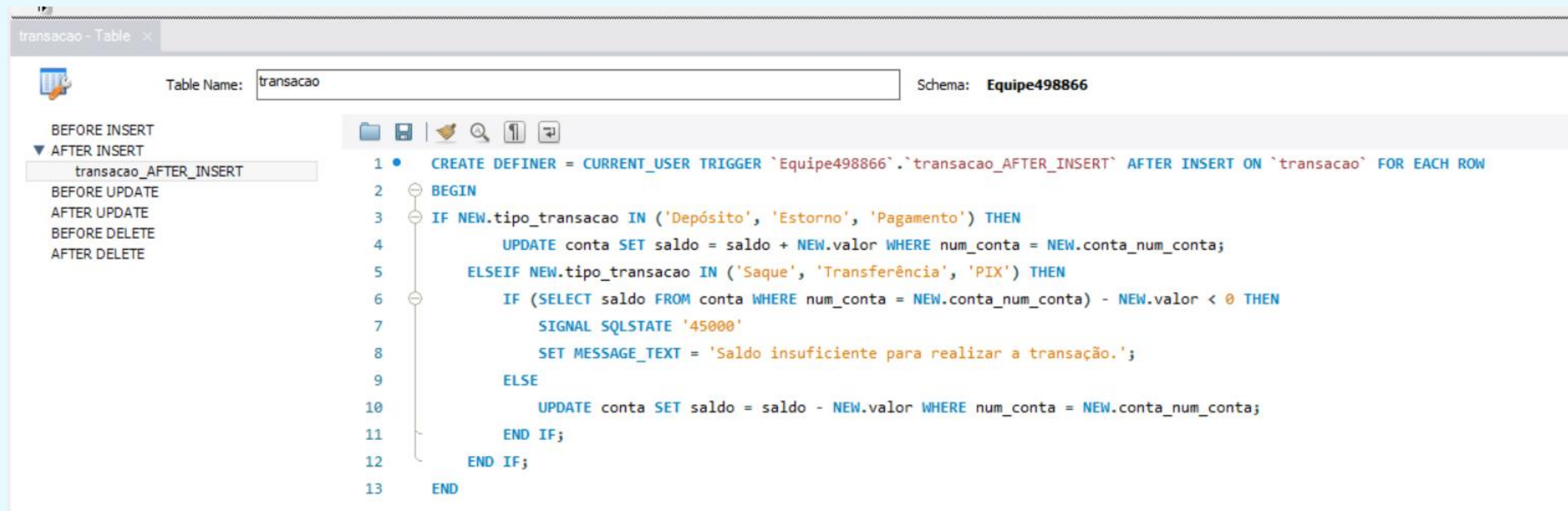


The screenshot shows a database IDE window titled "dependente - Table". The "Table Name" field contains "dependente" and the "Schema" is "Equipe498866". On the left, a tree view shows the trigger "dependente\_BEFORE\_INSERT" selected under the "BEFORE INSERT" category. The main editor displays the SQL code for creating this trigger:

```
1 • CREATE DEFINER = CURRENT_USER TRIGGER `Equipe498866`.`dependente_BEFORE_INSERT` BEFORE INSERT ON `dependente` FOR EACH ROW
2 BEGIN
3     DECLARE num_dependentes INT;
4     DECLARE idade_dependente INT;
5
6     -- Conta o número de dependentes já associados ao funcionário
7     SELECT COUNT(*) INTO num_dependentes
8     FROM dependente
9     WHERE matricula = NEW.matricula;
10
11    -- Se o número de dependentes for 5 ou mais, gera um erro
12    IF num_dependentes > 5 THEN
13        SIGNAL SQLSTATE '45000'
14        SET MESSAGE_TEXT = 'O funcionário não pode ter mais do que 5 dependentes.';
15    END IF;
16
17    -- Calcula a idade do dependente
18    SET idade_dependente = TIMESTAMPDIFF(YEAR, NEW.data_nas, CURDATE());
19
20    -- Atualiza o valor da idade no novo dependente
21    SET NEW.idade = idade_dependente;
22 END
```



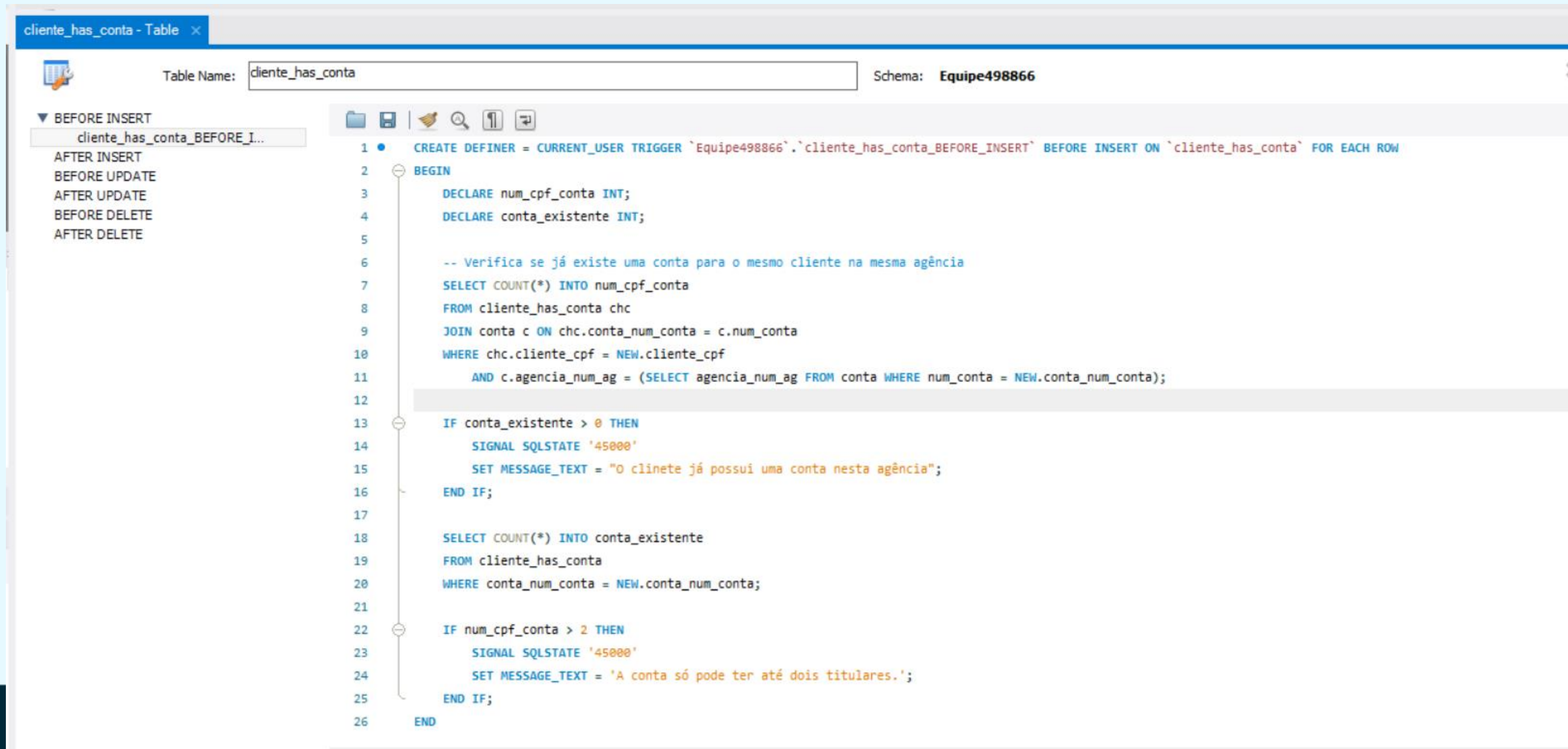
# Triggers – Transação



The screenshot shows a database IDE window titled "transacao - Table". The "Table Name" field contains "transacao" and the "Schema" is "Equipe498866". On the left, a tree view shows the trigger "transacao\_AFTER\_INSERT" selected under the "AFTER INSERT" category. The main editor displays the following SQL code:

```
1 CREATE DEFINER = CURRENT_USER TRIGGER `Equipe498866`.`transacao_AFTER_INSERT` AFTER INSERT ON `transacao` FOR EACH ROW
2 BEGIN
3 IF NEW.tipo_transacao IN ('Depósito', 'Estorno', 'Pagamento') THEN
4     UPDATE conta SET saldo = saldo + NEW.valor WHERE num_conta = NEW.conta_num_conta;
5 ELSEIF NEW.tipo_transacao IN ('Saque', 'Transferência', 'PIX') THEN
6     IF (SELECT saldo FROM conta WHERE num_conta = NEW.conta_num_conta) - NEW.valor < 0 THEN
7         SIGNAL SQLSTATE '45000'
8         SET MESSAGE_TEXT = 'Saldo insuficiente para realizar a transação.';
9     ELSE
10        UPDATE conta SET saldo = saldo - NEW.valor WHERE num_conta = NEW.conta_num_conta;
11    END IF;
12 END IF;
13 END
```

# Triggers – Cliente\_has\_conta



The screenshot shows a database IDE window titled 'cliente\_has\_conta - Table'. The 'Table Name' is 'cliente\_has\_conta' and the 'Schema' is 'Equipe498866'. On the left, a tree view shows the trigger 'cliente\_has\_conta\_BEFORE\_I...' under the 'BEFORE INSERT' category. The main editor displays the SQL code for creating this trigger.

```
1 CREATE DEFINER = CURRENT_USER TRIGGER `Equipe498866`.`cliente_has_conta_BEFORE_INSERT` BEFORE INSERT ON `cliente_has_conta` FOR EACH ROW
2 BEGIN
3     DECLARE num_cpf_conta INT;
4     DECLARE conta_existente INT;
5
6     -- Verifica se já existe uma conta para o mesmo cliente na mesma agência
7     SELECT COUNT(*) INTO num_cpf_conta
8     FROM cliente_has_conta chc
9     JOIN conta c ON chc.conta_num_conta = c.num_conta
10    WHERE chc.cliente_cpf = NEW.cliente_cpf
11           AND c.agencia_num_ag = (SELECT agencia_num_ag FROM conta WHERE num_conta = NEW.conta_num_conta);
12
13    IF conta_existente > 0 THEN
14        SIGNAL SQLSTATE '45000'
15        SET MESSAGE_TEXT = "O cliente já possui uma conta nesta agência";
16    END IF;
17
18    SELECT COUNT(*) INTO conta_existente
19    FROM cliente_has_conta
20    WHERE conta_num_conta = NEW.conta_num_conta;
21
22    IF num_cpf_conta > 2 THEN
23        SIGNAL SQLSTATE '45000'
24        SET MESSAGE_TEXT = "A conta só pode ter até dois titulares.";
25    END IF;
26 END
```



UFC70



UFC