**FACULDADES INTEGRADAS DE ARARAQUARA**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**LUIS FILIPE CAMBIAGHE GUEDES**

**PROJETO DE BANCO DE DADOS**

**TRABALHO 3**

**Prof. Dr. Fábio Papini Fornazari**

**ARARAQUARA – SP.**

**2018**

Sumário

[1. Descrição do problema 3](#_Toc516076425)

[2. Modelo Entidade Relacionamento (MER): 3](#_Toc516076426)

[3. Modelo relacional: 4](#_Toc516076427)

[4. Criação das tabelas ou “create table”: 4](#_Toc516076428)

[5. Função: 6](#_Toc516076429)

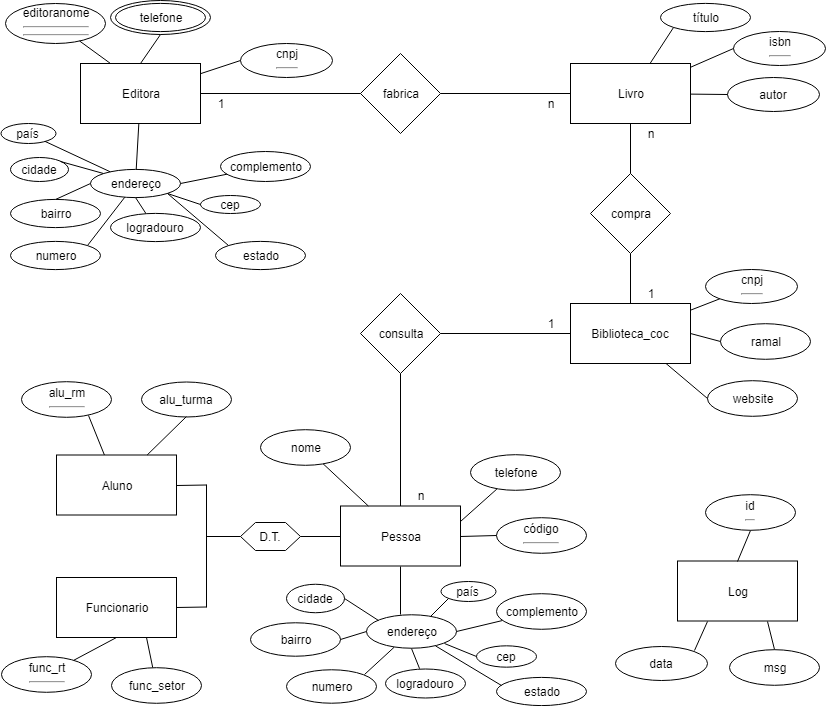
[6. Triggers 6](#_Toc516076430)

# Descrição do problema

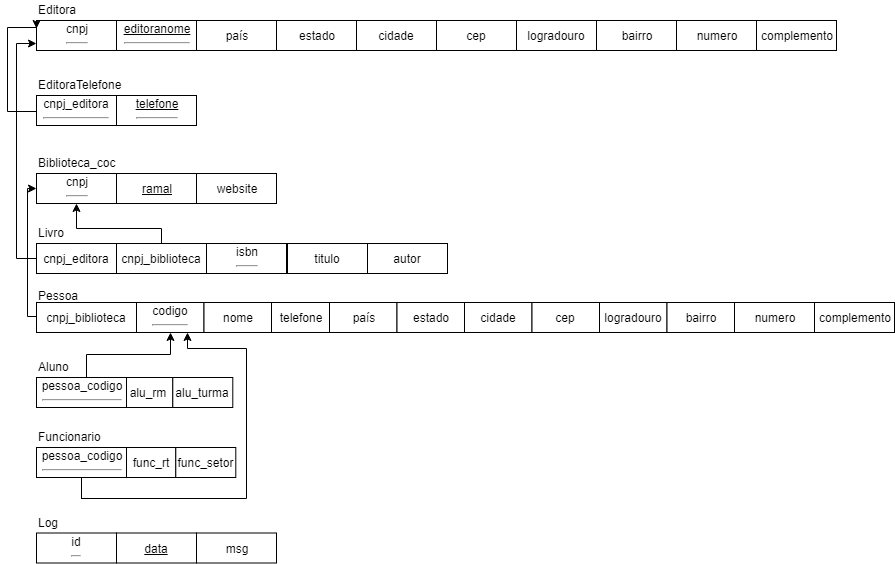
Deseja-se desenvolver o esquema do banco de dados de informações para uma única biblioteca de uma escola atendendo os requisitos:

* Manter dados sobre a Biblioteca da escola (CNPJ, RAMAL e WEBSITE)
* Para cada editora informações que ajudem na sua localização, contato e seu nome.
* Para cada livro manter seus títulos, código para controle e
* Manter as pessoas para que o bibliotecário (a) possa-os localizar facilmente dentro do colégio facilitando na entrega de livros e sugestão dos mesmos.
* É preciso manter um controle de tudo que é feito no cadastro das editoras e dos livros por segurança.

# Modelo Entidade Relacionamento (MER):



# Modelo relacional:



# Criação das tabelas ou “create table”:

CREATE TABLE Editora (

    cnpj varchar (100) not null,

    editoranome varchar (100) not null unique,

    pais varchar (100) not null,

estado varchar (100) not null,

cidade varchar (100) not null,

cep varchar (100) not null,

    logradouro varchar (100) not null,

bairro varchar (100) not null,

    numero int not null,

complemento varchar (100),

constraint PK\_Editora\_Cnpj primary Key (cnpj),

constraint UNQ\_EditoraNome unique (editoranome),

);

Create TABLE EditoraTelefone(

    cnpj\_editora varchar (100) not null,

    telefone varchar (20) not null,

constraint FK\_CnpjEditora foreign key (cnpj\_editora) references Editora (cnpj),

constraint PK\_EditoraTelefone\_CnpjEditora primary Key (telefone),

);

Create TABLE Biblioteca\_Coc(

    cnpj varchar (100) not null,

    ramal int not null,

    website varchar (100) not null,

    constraint PK\_Biblioteca\_Coc\_Cnpj primary Key (cnpj),

constraint UNQ\_BibliotecaRamal unique (ramal),

);

Create TABLE Livro(

cnpj\_editora varchar (100) not null,

cnpj\_biblioteca varchar (100) not null,

    isbn varchar (100) not null,

    qtde int not null,

titulo varchar (100) not null,

autor varchar (100) not null,

constraint FK\_Livro\_CnpjLivro foreign key (cnpj\_editora) references Editora (cnpj),

constraint FK\_Livro\_CnpjBiblioteca foreign key (cnpj\_biblioteca) references Biblioteca\_Coc (cnpj),

constraint PK\_Livro primary Key (cnpj\_editora, isbn),

);

Create TABLE Pessoa(

cnpj\_biblioteca varchar (100) not null,

codigo int IDENTITY(1,1) not null,

nome varchar (100) not null,

    telefone varchar (20) not null,

    pais varchar (100) not null,

estado varchar (100) not null,

cidade varchar (100) not null,

cep varchar (100) not null,

    logradouro varchar (100) not null,

bairro varchar (100) not null,

    numero int not null,

complemento varchar (100),

constraint FK\_Pessoa\_CnpjBiblioteca foreign key (cnpj\_biblioteca) references Biblioteca\_Coc (cnpj),

    constraint PK\_pessoa\_Codigo primary key (codigo),

);

Create TABLE Aluno(

pessoa\_codigo int,

alu\_rm int,

alu\_turma varchar (100),

constraint FK\_Aluno\_CodigoPessoa foreign key (pessoa\_codigo) references Pessoa (codigo),

    constraint PK\_Aluno\_AluRm primary key (alu\_rm),

);

Create TABLE Funcionario(

pessoa\_codigo int,

func\_rt int,

func\_setor varchar (100),

constraint FK\_Funcionario\_CodigoPessoa foreign key (pessoa\_codigo) references pessoa (codigo),

    constraint PK\_Funcionario\_FuncRt primary key (func\_rt),

);

CREATE TABLE Log

(

id int identity(1,1) NOT NULL,

data datetime,

msg varchar(max),

Constraint PK\_Log\_Id PRIMARY KEY(id)

);

# Função:

CREATE FUNCTION fnGetFoneEditora

(

@CNPJ varchar(20)

)

RETURNS VARCHAR(MAX)

AS

BEGIN

DECLARE @Result VARCHAR(MAX)=''

SELECT @Result = Coalesce(@Result + telefone + ', ', '')

FROM editoratelefone

WHERE cnpj\_editora = @CNPJ

SET @Result = SUBSTRING(@Result, 1, LEN(@Result)-1)

RETURN @Result

END

GO

# Triggers

-- Trigger LogLivro

IF EXISTS (SELECT name FROM sys.triggers WHERE name = 'TG\_LogLivro')

DROP TRIGGER TG\_LogLivro

GO

CREATE TRIGGER TG\_LogLivro

ON Livro

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

--Registar log

IF((SELECT count(\*) FROM INSERTED) > 0 AND (SELECT COUNT(\*) FROM DELETED)=0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Livro Inserido. ||CNPJ: ' + cnpj\_biblioteca

+ ' ||isbn: ' + isbn

+ ' ||qtde: ' + Convert(varchar(6),qtde)

+ ' ||Titulo: ' + titulo

+ ' ||autor: ' + autor

FROM INSERTED

END

ELSE IF((SELECT count(\*) FROM INSERTED)>0 AND (SELECT COUNT(\*) FROM DELETED)>0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Livro Alterado. ||CNPJ: ' + cnpj\_biblioteca

+ ' ||isbn: ' + isbn

+ ' ||qtde: ' + Convert(varchar(6),qtde)

+ ' ||Titulo: ' + titulo

+ ' ||autor: ' + autor

FROM INSERTED

END

ELSE IF ((SELECT COUNT(\*) FROM DELETED)>0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Livro Excluido. ||CNPJ: ' + cnpj\_biblioteca

+ ' ||isbn: ' + isbn

+ ' ||qtde: ' + Convert(varchar(6),qtde)

+ ' ||Titulo: ' + titulo

+ ' ||autor: ' + autor

FROM DELETED

END

END

GO

-- Trigger LogEditora

IF EXISTS (SELECT name FROM sys.triggers WHERE name = 'TG\_LogEditora')

DROP TRIGGER TG\_LogEditora

GO

CREATE TRIGGER TG\_LogEditora

ON Editora

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

--Registar log

IF((SELECT count(\*) FROM INSERTED) > 0 AND (SELECT COUNT(\*) FROM DELETED)=0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Editora Inserido. ||CNPJ: ' + cnpj

+ ' ||Nome da Editora: ' + editoranome

+ ' ||Pais: ' + pais

+ ' ||Estado: ' + estado

+ ' ||Cidade: ' + cidade

+ ' ||CEP: ' + cep

+ ' ||Logadrouro: ' + logradouro

+ ' ||Bairro: ' + bairro

+ ' ||Numero: ' + Convert(varchar(6),numero)

+ ' ||Complemento: ' + complemento

FROM INSERTED

END

ELSE IF((SELECT count(\*) FROM INSERTED)>0 AND (SELECT COUNT(\*) FROM DELETED)>0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Editora Alterado. ||CNPJ: ' + cnpj

+ ' ||Nome da Editora: ' + editoranome

+ ' ||Pais: ' + pais

+ ' ||Estado: ' + estado

+ ' ||Cidade: ' + cidade

+ ' ||CEP: ' + cep

+ ' ||Logadrouro: ' + logradouro

+ ' ||Bairro: ' + bairro

+ ' ||Numero: ' + Convert(varchar(6),numero)

+ ' ||Complemento: ' + complemento

FROM INSERTED

END

ELSE IF ((SELECT COUNT(\*) FROM DELETED)>0)

BEGIN

INSERT INTO Log(data, msg)

SELECT getdate()

, 'Registro de Editora Deletado. ||CNPJ: ' + cnpj

+ ' ||Nome da Editora: ' + editoranome

+ ' ||Pais: ' + pais

+ ' ||Estado: ' + estado

+ ' ||Cidade: ' + cidade

+ ' ||CEP: ' + cep

+ ' ||Logadrouro: ' + logradouro

+ ' ||Bairro: ' + bairro

+ ' ||Numero: ' + Convert(varchar(6),numero)

+ ' ||Complemento: ' + complemento

FROM INSERTED

END

END

GO

# Consulta:

-- 1.Consultar registro de um livro dado seu nome.

SELECT titulo as Título\_do\_Livro

, qtde as Quantidade\_do\_Livro

, autor as Autor\_do\_Livro

, biblioteca\_coc.website as Site\_da\_Biblio

, e.editoranome as Nome\_da\_Editora

, (select dbo.fnGetFoneEditora(e.CNPJ)) as Telefone\_da\_Editora

FROM livro

INNER JOIN editora e on cnpj\_editora = e.cnpj

INNER JOIN biblioteca\_coc on biblioteca\_coc.cnpj = biblioteca\_coc.cnpj

where titulo like '%Introdução a sistemas de bancos de dados%'

-- 2.Quantidade total de pessoas cadastradas

SELECT COUNT (codigo) as Numero\_de\_Pessoas\_Cadastradas from Pessoa

-- 3. Mostrar Log do sistemas passando uma certa data e hora:

SELECT data as Data\_do\_Ocorrido, msg as Ocorrencia from log

where data between '2001-06-06 00:00:00' and '2018-06-06 23:59:59'

-- 4. Mostrar dados para contato e endereço de uma editora dado seu nome

SELECT editoranome as Nome\_da\_editora

, (select dbo.fnGetFoneEditora(editora.CNPJ)) as Telefone\_da\_Editora

, estado as Estado

, cidade as Cidade

, cep as CEP

, logradouro as Endereco

, numero as Numero

from Editora

order by editoranome

where editoranome = 'Editora Abril'

-- 5. Mostrar dados de um aluno dado registro.

SELECT pessoa.nome as Nome\_da\_Pessoa

, pessoa.telefone as Telefone\_da\_Pessoa

, pessoa.logradouro as Endereco\_da\_Pessoa

, pessoa.numero as NumeroResidencial

, alu\_turma as Turma\_do\_Aluno

from aluno

INNER JOIN pessoa on pessoa\_codigo = pessoa.codigo

    where alu\_rm = 44325

-- 6. Mostrar dados de um aluno dado registro.

SELECT pessoa.nome as Nome\_da\_Pessoa

, pessoa.telefone as Telefone\_da\_Pessoa

, pessoa.logradouro as Endereco\_da\_Pessoa

, pessoa.numero as NumeroResidencial

, func\_setor as Setor\_do\_Funcionario

from funcionario

INNER JOIN pessoa on pessoa\_codigo = pessoa.codigo

where func\_rt = 05140055