

**Universidade Federal de Mato Grosso do Sul  
Faculdade de Computação  
Banco de Dados - T01 - Prof. Hercules Sandim**

**TRABALHO PRÁTICO  
Projeto - Cartório de Protestos**

Matheus Lima Machado (201519060068)

Nathaly Veneruchi (201519060424)

Campo Grande, MS

2017

## Introdução

Neste trabalho foi desenvolvido um projeto de um sistema de banco de dados de um Cartório - 3º Tabelionato de Protesto. Um cartório de protesto é uma instituição pública onde o credor apresenta títulos para serem protestados. O protesto de títulos é um direito do credor para provar a obrigação que tinha o devedor de lhe pagar uma quantia certa, reconhecida como dívida em dinheiro, que não foi cumprida na data estabelecida para pagamento. Com o protesto, o credor comprova a mora (atraso da obrigação) e a inadimplência (descumprimento da obrigação) do devedor.

Depois de receber os títulos ou documentos de dívidas, o Tabelião os protocola, em ordem seqüencial crescente, emite intimação para o devedor pagar o que deve no prazo de três dias úteis em que o título foi protestado.

Se o devedor for desconhecido no endereço fornecido pelo credor; se residir em local fora da competência territorial do Tabelionato; se ninguém se dispuser a receber a intimação; ou então se a sua localização for incerta ou ignorada, o Tabelião, após receber a informação por parte de quem levou a intimação ao devedor, ou se o credor já requerer no ato da apresentação, a intimação direta por edital, os dados do devedor, do título e do credor são publicados em edital, que tem legalmente o mesmo efeito da intimação feita diretamente à pessoa. Os editais são publicados em jornal de grande circulação diária.

Após receberem as intimações ou terem-nas publicadas na imprensa, os devedores têm três dias úteis, não contando o do recebimento, para efetuar o pagamento, o aceite ou a devolução, ou procurarem o credor para negociar ou negar a dívida, ou interpor ação de sustação de protesto. Podem também encaminhar ao Tabelionato as razões de porque não efetuará o pagamento, o aceite ou a devolução, mas esta providência não impede o protesto.

Quando o devedor paga, o faz pelo valor do título, acrescido dos emolumentos e custos incorridos pelo Tabelionato e Ofício de Distribuição. O valor referente ao título é repassado para o credor no primeiro dia útil seguinte e a dívida considera-se quitada. Deve o devedor passar no Tabelionato e pegar o título que foi levado a protesto, para sua garantia.

A data do protesto serve para indicar o dia inicial em que se contarão juros, taxas e correções monetárias sobre a dívida, se estes não estiverem já estipulados em um contrato.

## **Descrição do minimundo**

O Cartório possui colaboradores, que foram chamados de Funcionários no projeto. Em ordem hierárquica, estas são as subclasses de funcionários:

*Tabelião* - Dono do cartório. Pode atender pedidos de usuários;

*Tabelião substituto* - Atua como dono do cartório caso o Tabelião não esteja presente.

*Escreventes* - Podem assinar documentos importantes. Podem atender pedidos de usuários (tipos: Master1 > Master2 > Sênior1 > Sênior2)

*Auxiliares* - Podem assinar documentos importantes. Podem atender pedidos de usuários. Podem operar o caixa . Podem auxiliar seus superiores.

*Auxiliar geral* - Responsáveis por serviços gerais.

O Cartório atende usuários, estes podem ser uma pessoa física ou uma pessoa jurídica. Um usuário pode ser credor, devedor ou apresentante de/em um certo título. Um devedor é um usuário que deve uma quantia (valor) em dinheiro para um credor. O apresentante é o usuário que apresenta o título cobrando essa dívida.

Na maioria das vezes quem apresenta os títulos são os Bancos (que são pessoas jurídicas), essas empresas possuem convênio com os credores e enviam, através de uma única Central de Remessa de Arquivos (CRA), os títulos para o Cartório.

Se estiver regular, o título possui um prazo de 3 dias e caso não seja pago, será protestado. Antes de ser protestado, o título pode ser retirado pelo apresentante; pode ser cancelado pelo devedor, se este possuir os documentos necessários. Após protestado, o título pode ser sustado ou suspenso por uma determinação em um ofício vindo de um Tribunal. O Cartório, envia um “Ofício de Resposta” para o tribunal, para informar que executou as determinações do Ofício recebido.

Quando um usuário (requerente), faz um requerimento de título(s). É iniciado um processo de cancelamento, onde ficam armazenados todos os títulos que foi solicitado o cancelamento, e a certidão de cancelamento de cada título.

Um usuário pode solicitar uma certidão de protesto para o Cartório, que considera uma certidão “positiva” aquela que possui títulos, e “negativa” uma que não possui título algum. Em uma certidão de protesto positiva consta os títulos de um determinado devedor.

### **Entidades identificadas:**

Usuário (id, nome\_completo, dt\_nasc, tel, email, CEP, rua, numero, bairro, cidade)

Usuário (id, nome\_completo, dt\_nasc, tel, email, CEP, rua, numero, bairro, cidade)

Funcionário (cpf, rg, sexo, estado\_civil, data\_contratacao, nome\_completo, rua, numero, bairro, cidade, CEP, tel, email, ctps, função)

Pedido (protocolo\_pedido, tipo, data\_entrada, requerente, usuário, func\_responsavel)

Tribunal (id, nome, cidade, estado, tipo)

Ofício(id\_oficio, autos, data\_recepção, protocolo\_titulo, tribunal\_id)

Ofício\_de\_Resposta (nro\_id, resposta, data\_emissao, id\_oficio)

Instrumento\_de\_protesto (livro\_folha, tipo\_protesto, notificação, data\_do\_protesto, valor\_protestado, protocolo\_titulo)

Proceso\_de\_Cancelamento (nro\_processo, protocolo\_pedido, )

Certidao\_Cancelamento(selo, livro\_folha, nro\_processo)

Certidao\_Protesto(selo, protocolo\_pedido)

### **Especializações da entidade Usuário:**

Pessoa\_Física (id\_usuario, sexo, cpf, rg)

Pessoa\_Jurídica (id\_usuario, cnpj)

### **Especializações da entidade Funcionário:**

Escrevente (cpf\_funcionario, tipo)

Auxiliar (cpf\_funcionario, setor)

## Esclarecimentos sobre entidades

Funcionário (cpf, nome\_completo, rg, sexo, tel, email, endereco, ctps, pis, salario, data\_contratacao, estado-civil, função)

- O atributo *ctps* se refere ao número da carteira de trabalho do funcionário.
- O atributo *pis* se refere ao número do PIS (Programa de Integração Social) do funcionário.

Título (protocolo\_titulo, nro\_titulo, valor, vencimento, data\_emissão, credor, devedor, apresentante, espécie, endosso, aceite, ocorrencias)

- O atributo *nro\_titulo* é um código de identificação gerado pelo Cartório Distribuidor (responsável por receber os títulos do CRA e encaminhar para os demais cartórios). É um jeito que o Cartório Distribuidor identifica os títulos.
- O atributo *protocolo\_titulo* é um código de identificação gerado pelo Cartório - 3º Tabelionato de Protestos. É um jeito próprio que o este cartório identifica os títulos.
- O atributo *espécie* pode ser: duplicata mercantil (DM); cheque; nota promissória; duplicata mercantil por indicação (DMI) (quando é apresentada por um banco); certidão de dívida ativa (CDA).
- O atributo *endosso* pode ser: translativo; mandatário; null.
- O atributo  *aceite*: significado com ou sem aceite.
- O atributo *ocorrencias* pode ser: irregular(antes de ser protestado); protestado; cancelado (depois de ser protestado); suspenso (depois de protestado); pago (antes de ser protestado); sustado (antes de protestado); retirado (antes de protestado).

Instrumento\_de\_protesto (livro\_folha, tipo\_protesto, notificação, data\_do\_protesto, valor\_protestado, protocolo\_titulo)

- O atributo *notificação* pode ser: carta AR (aviso recebido); edital.
- O atributo *tipo\_protesto* pode ser: comum; falimentar.

Ofício\_de\_Resposta (nro\_id, protocolo\_oficio, resposta, data\_emissao)

- O atributo *resposta* é um texto que descreve o que foi feito.

Ofício(id\_oficio, tribunal, data\_recepção, autos, protocolo\_titulo)

- O atributo autos é um código de identificação de um processo, é usado pelo tribunal.

Tribunal (id, nome, cidade, estado, tipo)

- O atributo *tipo* pode ser: federal, estadual.

Pedido (protocolo\_pedido, requerente, usuário, tipo, data\_entrada, livro\_folha)

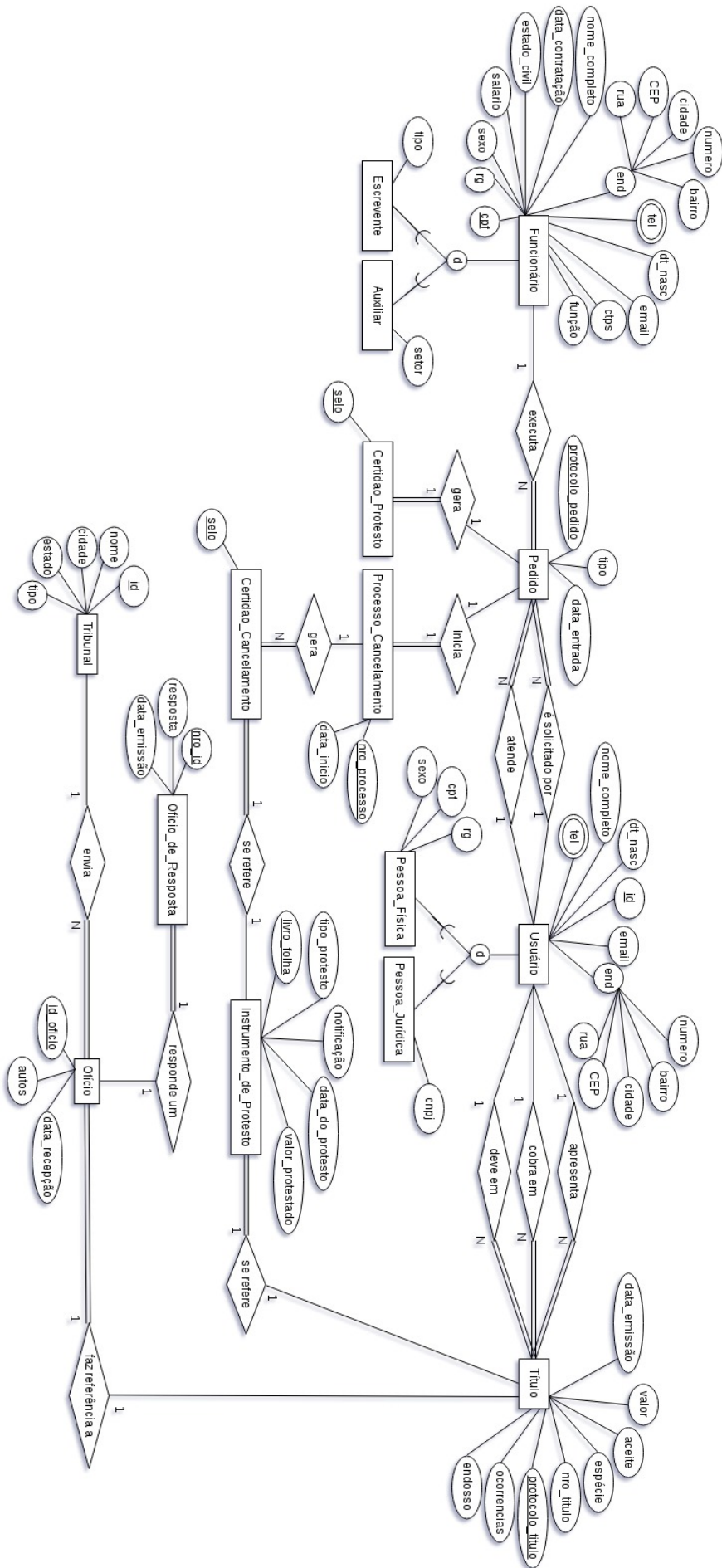
- O atributo *tipo* pode ser: certidão, cancelamento.

Certidao\_Cancelamento(selo, livro\_folha)

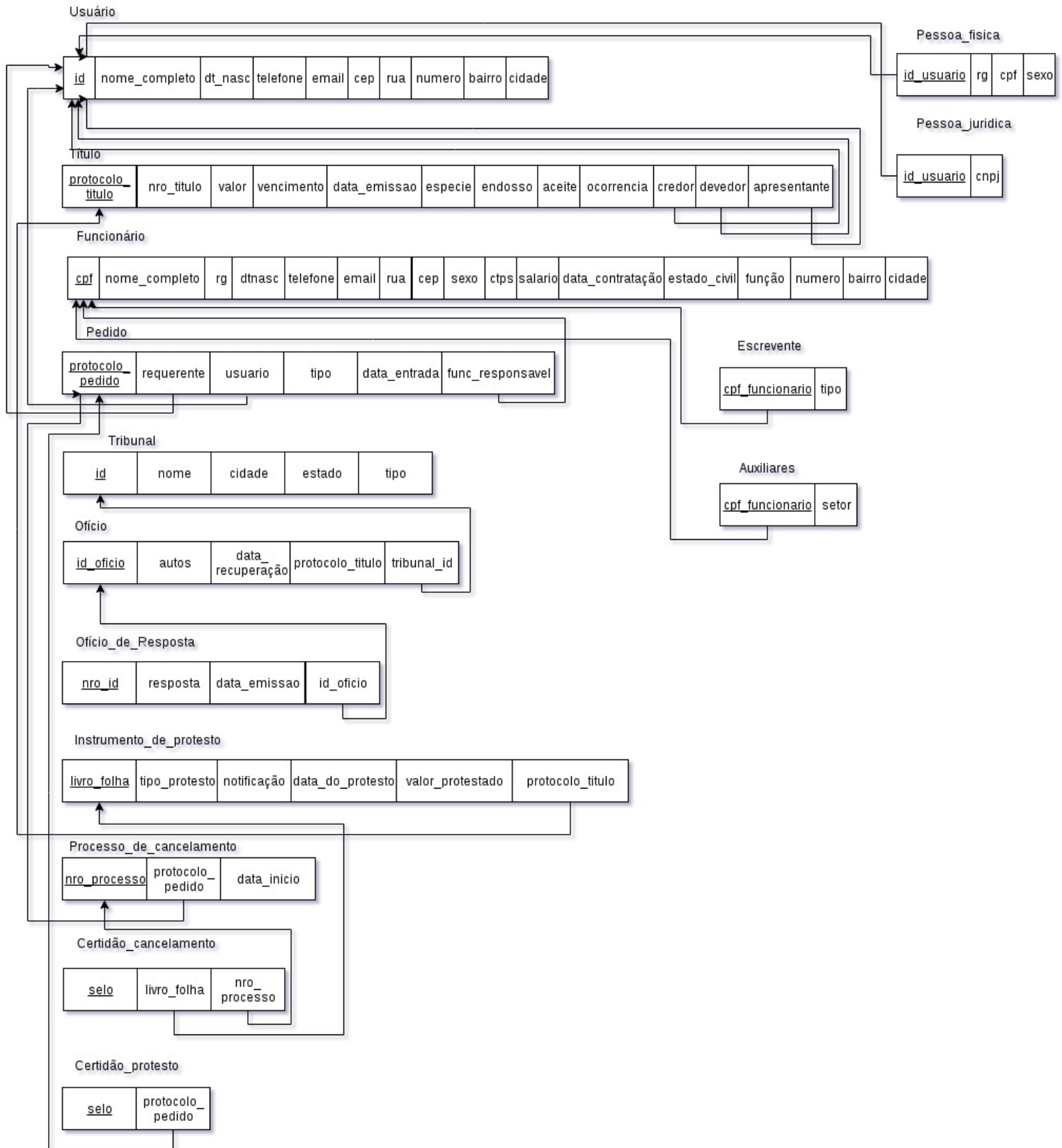
Certidao\_Protesto(selo, protocolo\_pedido)

- O atributo *selo* dessas duas entidades é usado pela corregedoria, um órgão regulador dos cartórios.

# Modelo Conceitual - Diagrama E-R



# Modelo Relacional





## Modelagem Física - MySQL

```
CREATE TABLE usuario(  
    id INTEGER NOT NULL,  
    nome_completo VARCHAR(50) NOT NULL,  
    dt_nasc DATE NOT NULL,  
    tel INTEGER,  
    email VARCHAR(50),  
    rg INTEGER NOT NULL,  
    cep INTEGER NOT NULL,  
    rua VARCHAR(80) NOT NULL,  
    numero INTEGER NOT NULL,  
    bairro VARCHAR(30) NOT NULL,  
    cidade VARCHAR(30) NOT NULL,  
  
    PRIMARY KEY (id)  
  
);
```

```
CREATE TABLE funcionario(  
    cpf char(11) NOT NULL,  
    rg INTEGER NOT NULL,  
    sexo ENUM('M', 'F') NOT NULL,  
    salario INTEGER NOT NULL,  
    estado_civil VARCHAR(10),  
    data_contratacao date NOT NULL,  
    nome_completo VARCHAR(30) NOT NULL,  
    email VARCHAR(30),  
    rua VARCHAR(130) NOT NULL,  
    numero INTEGER NOT NULL,  
    bairro VARCHAR(30) NOT NULL,  
    cidade VARCHAR(30) NOT NULL,  
    cep char(8) NOT NULL,  
    tel INTEGER NOT NULL,  
    dt_nasc DATE NOT NULL,  
    ctps INTEGER NOT NULL,  
    funcao VARCHAR(30) NOT NULL,  
  
    PRIMARY KEY (cpf)  
  
);
```

```

CREATE TABLE titulo(
  protocolo_titulo INTEGER NOT NULL,
  ocorrencia VARCHAR(30) NOT NULL,
  endosso VARCHAR(30) NOT NULL,
  nro_titulo INTEGER NOT NULL,
  especie VARCHAR(30),
  aceite BOOLEAN NOT NULL,
  valor INTEGER NOT NULL,
  data_emissao date NOT NULL,
  credor INTEGER NOT NULL,
  devedor INTEGER NOT NULL,
  apresentante INTEGER NOT NULL,

  PRIMARY KEY (protocolo_titulo),
  FOREIGN KEY (credor) REFERENCES usuario(id),
  FOREIGN KEY (devedor) REFERENCES usuario(id),
  FOREIGN KEY (apresentante) REFERENCES usuario(id)
);

```

```

CREATE TABLE instrumento_de_protesto(
  livro_folha VARCHAR(30) NOT NULL,
  tipo_protesto VARCHAR(30) NOT NULL,
  notificacao VARCHAR(30) NOT NULL,
  data_do_protesto DATE NOT NULL,
  valor_protestado DOUBLE NOT NULL,
  protocolo_titulo INTEGER NOT NULL,

  PRIMARY KEY (livro_folha),
  FOREIGN KEY (protocolo_titulo) REFERENCES titulo(protocolo_titulo)
);

```

```

CREATE TABLE tribunal (
  id INTEGER NOT NULL,
  nome VARCHAR (30) NOT NULL,
  cidade VARCHAR (30) NOT NULL,
  estado VARCHAR (30) NOT NULL,
  tipo VARCHAR (30) NOT NULL,

  PRIMARY KEY (id)

```

```

);

```

```

CREATE TABLE oficio (
id_oficio INTEGER NOT NULL,
data_recepção date NOT NULL,
autos INTEGER NOT NULL,
protocolo_titulo INTEGER NOT NULL,
tribunal_id INTEGER NOT NULL,

PRIMARY KEY (id_oficio),
FOREIGN KEY (protocolo_titulo) REFERENCES titulo (protocolo_titulo),
FOREIGN KEY (tribunal_id) REFERENCES tribunal (id)
);

```

```

CREATE TABLE oficio_de_resposta (
nro_id INTEGER NOT NULL,
resposta VARCHAR(30) NOT NULL,
data_emissao date NOT NULL,
id_oficio INTEGER NOT NULL,

PRIMARY KEY (nro_id),
FOREIGN KEY (id_oficio) REFERENCES oficio (id_oficio)
);

```

```

CREATE TABLE pedido (
protocolo_pedido INTEGER NOT NULL,
tipo VARCHAR (30) NOT NULL,
data_entrada date NOT NULL,
usuario INTEGER NOT NULL,
requerente INTEGER NOT NULL,
func_responsavel CHAR(11) NOT NULL,

PRIMARY KEY (protocolo_pedido),
FOREIGN KEY (requerente) REFERENCES usuario (id),
FOREIGN KEY (usuario) REFERENCES usuario(id),
FOREIGN KEY (func_responsavel) REFERENCES funcionario (cpf)
);

```

```

CREATE TABLE certidao_cancelamento(
selo_cc VARCHAR (30) NOT NULL,
livro_folha VARCHAR (30) NOT NULL,
nro_processo INTEGER NOT NULL,

PRIMARY KEY (selo_cc),
FOREIGN KEY (livro_folha) REFERENCES instrumento_de_protesto
(livro_folha),
FOREIGN KEY (nro_processo) REFERENCES processo_de_cancelamento
(nro_processo)

);

```

```

CREATE TABLE certidao_protesto (
selo_cp VARCHAR (30) NOT NULL,
protocolo_pedido INTEGER NOT NULL,

PRIMARY KEY (selo_cp),
FOREIGN KEY (protocolo_pedido) REFERENCES pedido(protocolo_pedido)
);

```

```

CREATE TABLE processo_de_cancelamento(
nro_processo INTEGER NOT NULL,
requerente INTEGER NOT NULL,
livro_folha VARCHAR (30) NOT NULL,
protocolo_pedido INTEGER NOT NULL,
data_inicio DATE NOT NULL,

PRIMARY KEY (nro_processo),
FOREIGN KEY (protocolo_pedido) REFERENCES pedido (protocolo_pedido)
);

```

```
CREATE TABLE escrevente (  
  cpf_funcionario char(11)NOT NULL,  
  nome_completo VARCHAR (30) NOT NULL,  
  rg char(9)NOT NULL ,  
  sexo enum('M', 'F') NOT NULL,  
  tel char(9)NOT NULL,  
  email VARCHAR (30) NOT NULL,  
  endereco VARCHAR (30) NOT NULL,  
  ctps INTEGER NOT NULL,  
  salario DOUBLE NOT NULL,  
  data_contratacao date NOT NULL,  
  estado_civil VARCHAR(10),  
  funcao VARCHAR(30) NOT NULL,  
  tipo VARCHAR(30) NOT NULL,
```

```
PRIMARY KEY (cpf_funcionario)
```

```
);
```

```
CREATE TABLE auxiliar (  
  cpf_funcionario char(11)NOT NULL,  
  nome_completo VARCHAR (30) NOT NULL,  
  rg char(9)NOT NULL ,  
  sexo enum('M', 'F') NOT NULL,  
  tel char(9)NOT NULL,  
  email VARCHAR (30) NOT NULL,  
  endereco VARCHAR (30) NOT NULL,  
  ctps INTEGER NOT NULL,  
  salario DOUBLE NOT NULL,  
  data_contratacao date NOT NULL,  
  estado_civil VARCHAR(10),  
  funcao VARCHAR(30) NOT NULL,  
  setor VARCHAR(30) NOT NULL,
```

```
PRIMARY KEY (cpf_funcionario)
```

```
);
```

```

CREATE TABLE pessoa_Física (
    id_usuario INTEGER NOT NULL,
    nome_completo VARCHAR(50) NOT NULL,
    dt_nasc DATE NOT NULL,
    tel INTEGER,
    email VARCHAR(50),
    rg INTEGER NOT NULL,
    cep INTEGER NOT NULL,
    rua VARCHAR(80) NOT NULL,
    numero INTEGER NOT NULL,
    bairro VARCHAR(30) NOT NULL,
    cidade VARCHAR(30) NOT NULL,
    sexo enum('M', 'F') NOT NULL,
    cpf char(11)NOT NULL,

    PRIMARY KEY (id_usuario)
);

CREATE TABLE pessoa_jurídica (
    id_usuario INTEGER NOT NULL,
    nome_completo VARCHAR(50) NOT NULL,
    dt_nasc DATE NOT NULL,
    tel INTEGER,
    email VARCHAR(50),
    cep INTEGER NOT NULL,
    rua VARCHAR(80) NOT NULL,
    numero INTEGER NOT NULL,
    bairro VARCHAR(30) NOT NULL,
    cidade VARCHAR(30) NOT NULL,
    cnpj char(11)NOT NULL,

    FOREIGN KEY (id_usuario) REFERENCES usuario(id)
);

```

```
CREATE TABLE usuario(  
  id INTEGER NOT NULL,  
  nome_completo VARCHAR(50) NOT NULL,  
  dt_nasc DATE NOT NULL,  
  tel INTEGER,  
  email VARCHAR(50),  
  rg INTEGER NOT NULL,  
  cep INTEGER NOT NULL,  
  rua VARCHAR(80) NOT NULL,  
  numero INTEGER NOT NULL,  
  bairro VARCHAR(30) NOT NULL,  
  cidade VARCHAR(30) NOT NULL,  
  
  PRIMARY KEY (id)  
  
);
```

```
CREATE TABLE funcionario(  
  cpf char(11)NOT NULL,  
  rg INTEGER NOT NULL,  
  sexo ENUM('M', 'F') NOT NULL,  
  salario INTEGER NOT NULL,  
  estado_civil VARCHAR(10),  
  data_contratacao date NOT NULL,  
  nome_completo VARCHAR(30) NOT NULL,  
  email VARCHAR(30),  
  rua VARCHAR(130) NOT NULL,  
  numero INTEGER NOT NULL,  
  bairro VARCHAR(30) NOT NULL,  
  cidade VARCHAR(30) NOT NULL,  
  cep char(8) NOT NULL,  
  tel INTEGER NOT NULL,  
  dt_nasc DATE NOT NULL,  
  ctps INTEGER NOT NULL,  
  funcao VARCHAR(30) NOT NULL,  
  
  PRIMARY KEY (cpf)  
  
);
```

```

CREATE TABLE titulo(
  protocolo_titulo INTEGER NOT NULL,
  ocorrencia VARCHAR(30)NOT NULL,
  endosso VARCHAR(30) NOT NULL,
  nro_titulo INTEGER NOT NULL,
  especie VARCHAR(30),
  aceite BOOLEAN NOT NULL,
  valor INTEGER NOT NULL,
  data_emissao date NOT NULL,
  credor INTEGER NOT NULL,
  devedor INTEGER NOT NULL,
  apresentante INTEGER NOT NULL,

  PRIMARY KEY (protocolo_titulo),
  FOREIGN KEY (credor) REFERENCES usuario(id),
  FOREIGN KEY (devedor) REFERENCES usuario(id),
  FOREIGN KEY (apresentante) REFERENCES usuario(id)
);

```

```

CREATE TABLE instrumento_de_protesto(
  livro_folha VARCHAR(30) NOT NULL,
  tipo_protesto VARCHAR(30) NOT NULL,
  notificacao VARCHAR(30) NOT NULL,
  data_do_protesto DATE NOT NULL,
  valor_protestado DOUBLE NOT NULL,
  protocolo_titulo INTEGER NOT NULL,

  PRIMARY KEY (livro_folha),
  FOREIGN KEY (protocolo_titulo) REFERENCES titulo(protocolo_titulo)
);

```

```

CREATE TABLE tribunal (
  id INTEGER NOT NULL,
  nome VARCHAR (30) NOT NULL,
  cidade VARCHAR (30) NOT NULL,
  estado VARCHAR (30) NOT NULL,
  tipo VARCHAR (30) NOT NULL,

  PRIMARY KEY (id)

);

```



```

CREATE TABLE oficio (
id_oficio INTEGER NOT NULL,
data_recepção date NOT NULL,
autos INTEGER NOT NULL,
protocolo_titulo INTEGER NOT NULL,
tribunal_id INTEGER NOT NULL,

PRIMARY KEY (id_oficio),
FOREIGN KEY (protocolo_titulo) REFERENCES titulo (protocolo_titulo),
FOREIGN KEY (tribunal_id) REFERENCES tribunal (id)
);

```

```

CREATE TABLE oficio_de_resposta (
nro_id INTEGER NOT NULL,
resposta VARCHAR(30) NOT NULL,
data_emissao date NOT NULL,
id_oficio INTEGER NOT NULL,

PRIMARY KEY (nro_id),
FOREIGN KEY (id_oficio) REFERENCES oficio (id_oficio)
);

```

```

CREATE TABLE pedido (
protocolo_pedido INTEGER NOT NULL,
tipo VARCHAR (30) NOT NULL,
data_entrada date NOT NULL,
usuario INTEGER NOT NULL,
requerente INTEGER NOT NULL,
func_responsavel CHAR(11) NOT NULL,

PRIMARY KEY (protocolo_pedido),
FOREIGN KEY (requerente) REFERENCES usuario (id),
FOREIGN KEY (usuario) REFERENCES usuario(id),
FOREIGN KEY (func_responsavel) REFERENCES funcionario (cpf)
);

```

```

CREATE TABLE certidao_cancelamento(
selo_cc VARCHAR (30) NOT NULL,
livro_folha VARCHAR (30) NOT NULL,
nro_processo INTEGER NOT NULL,

PRIMARY KEY (selo_cc),
FOREIGN KEY (livro_folha) REFERENCES instrumento_de_protesto
(livro_folha),
FOREIGN KEY (nro_processo) REFERENCES processo_de_cancelamento
(nro_processo)

```

```
);
```

```

CREATE TABLE certidao_protesto (
selo_cp VARCHAR (30) NOT NULL,
protocolo_pedido INTEGER NOT NULL,

```

```

PRIMARY KEY (selo_cp),
FOREIGN KEY (protocolo_pedido) REFERENCES pedido(protocolo_pedido)
);

```

```

CREATE TABLE processo_de_cancelamento(
nro_processo INTEGER NOT NULL,
requerente INTEGER NOT NULL,
livro_folha VARCHAR (30) NOT NULL,
protocolo_pedido INTEGER NOT NULL,
data_inicio DATE NOT NULL,

```

```

PRIMARY KEY (nro_processo),
FOREIGN KEY (protocolo_pedido) REFERENCES pedido (protocolo_pedido)
);

```

```

CREATE TABLE escrevente (
  cpf_funcionario char(11)NOT NULL,
  nome_completo VARCHAR (30) NOT NULL,
  rg char(9)NOT NULL ,
  sexo enum('M', 'F') NOT NULL,
  tel char(9)NOT NULL,
  email VARCHAR (30) NOT NULL,
  endereco VARCHAR (30) NOT NULL,
  ctps INTEGER NOT NULL,
  salario DOUBLE NOT NULL,
  data_contratacao date NOT NULL,
  estado_civil VARCHAR(10),
  funcao VARCHAR(30) NOT NULL,
  tipo VARCHAR(30) NOT NULL,

  PRIMARY KEY (cpf_funcionario),
  FOREIGN KEY (cpf_funcionario) REFERENCES funcionario(cpf)

);

```

```

CREATE TABLE auxiliar (
  cpf_funcionario char(11)NOT NULL,
  nome_completo VARCHAR (30) NOT NULL,
  rg char(9)NOT NULL ,
  sexo enum('M', 'F') NOT NULL,
  tel char(9)NOT NULL,
  email VARCHAR (30) NOT NULL,
  endereco VARCHAR (30) NOT NULL,
  ctps INTEGER NOT NULL,
  salario DOUBLE NOT NULL,
  data_contratacao date NOT NULL,
  estado_civil VARCHAR(10),
  funcao VARCHAR(30) NOT NULL,
  setor VARCHAR(30) NOT NULL,

  PRIMARY KEY (cpf_funcionario),
  FOREIGN KEY (cpf_funcionario) REFERENCES funcionario(cpf)

);

```

```

CREATE TABLE pessoa_Física (
    id_usuario INTEGER NOT NULL,
    nome_completo VARCHAR(50) NOT NULL,
    dt_nasc DATE NOT NULL,
    tel INTEGER,
    email VARCHAR(50),
    rg INTEGER NOT NULL,
    cep INTEGER NOT NULL,
    rua VARCHAR(80) NOT NULL,
    numero INTEGER NOT NULL,
    bairro VARCHAR(30) NOT NULL,
    cidade VARCHAR(30) NOT NULL,
    sexo enum('M', 'F') NOT NULL,
    cpf char(11)NOT NULL,

    PRIMARY KEY (id_usuario)
    FOREIGN KEY (id_usuario) REFERENCES usuario(id)

);

```

```

CREATE TABLE pessoa_jurídica (
    id_usuario INTEGER NOT NULL,
    nome_completo VARCHAR(50) NOT NULL,
    dt_nasc DATE NOT NULL,
    tel INTEGER,
    email VARCHAR(50),
    cep INTEGER NOT NULL,
    rua VARCHAR(80) NOT NULL,
    numero INTEGER NOT NULL,
    bairro VARCHAR(30) NOT NULL,
    cidade VARCHAR(30) NOT NULL,
    cnpj char(11)NOT NULL,

    PRIMARY KEY (id_usuario),
    FOREIGN KEY (id_usuario) REFERENCES usuario(id)

);

```

## Povoamento do Banco De Dados

### Povoando usuário:

```
INSERT INTO usuario VALUES (987, 'jose da silva', '1987/01/01',  
3333333, 'jose@dasilva.com', 999999, 8484884, 'rua joaquim',  
334, 'bairro', 'campo grande');
```

```
INSERT INTO usuario VALUES (456, 'joao antonio filho',  
'1986/02/04', 99999, 'joao@antonio.com', 888888,93839, 'rua  
ruarua', 456, 'bairrodois', 'campo grande');
```

```
INSERT INTO usuario VALUES (124, 'maria joaquina rosa',  
'1990/5/07', 8464688, 'joaquina@rosa.com', 0303030,93939, 'rua  
ruadois', 678, 'airrotres', 'campo grande');
```

```
INSERT INTO usuario VALUES (122, 'Flávio José', '1990/05/04',  
39464569, 'flavio@jose.com', 04645620,389383, 'ruatres', 789,  
'bairroquatro', 'campo grande');
```

### Povoando pessoa\_juridica:

```
INSERT INTO pessoa_jurídica VALUES ('933', 'banco do brasil',  
'1987/09/08', '93893', 'email', '98298', 'rua tal', '889',  
'bairro', 'cg', '28928');
```

```
INSERT INTO pessoa_jurídica VALUES ('409', 'caixa economica',  
'1986/09/08', '333223', 'email', '398938', 'rua', '928',  
'bairro', 'cg', '39893');
```

```
INSERT INTO pessoa_jurídica VALUES ('144', 'coca cola',  
'1987/01/02', '8982', 'email', '3839', 'ruaa', '3983', 'bairro',  
'cg', '39398');
```

```
INSERT INTO pessoa_jurídica VALUES ('123', 'juridica',  
'1990/08/07', '93893', 'email', '3938', 'rua um', '93893',  
'bairro tal', 'cg ', '93893');
```

### Povoando pessoa\_física:

```
INSERT INTO pessoa_física VALUES ('987', 'jose da silva',  
'1987/01/01', '3897237', 'jose@email.com', '5656', '3484', 'rua  
um', '3773', 'bairro um', 'campo grande ', 'M', '3988');
```

```
INSERT INTO pessoa_física VALUES ('456', 'joao antonioa',  
'1986/02/04', '9393939', 'email', '55389', '999', 'rua dois',  
'4545', 'bairro dois', 'CG', 'M', '777');
```

```
INSERT INTO pessoa_física VALUES ('124', 'maria', '9090/09/09',  
'38938983', 'email', '5693', '99', 'rua tres', '383', 'bairro  
tres', 'cg', 'F', '9999');
```

```
INSERT INTO pessoa_física VALUES ('122', 'flavio jose',
'1990/05/04', '938938', 'email', '3555', '8983', 'rua quatro',
'938', 'bairro quatro', 'cg', 'M', '3893');
```

#### **Povoando titulo:**

```
INSERT INTO titulo VALUES ('1', 'suspensao', 'translativo',
'454', 'DM', '1', '4554', '2017/03/02', '5345435', '56656',
'546545');
INSERT INTO titulo VALUES ('2', 'protestado', 'translativo',
'4566', 'DM', '1', '345', '2017/03/02', '45345', '5556',
'45345');
INSERT INTO titulo VALUES ('3', 'protestado', 'translativo',
'6776', 'DMI', '1', '455', '2017/03/02', '33456', '56666',
'345345');
INSERT INTO titulo VALUES ('4', 'irregular', 'mandatario',
'676', 'CDA', '1', '455', '2017/03/02', '6456', '6666',
'54555');
```

#### **Povoando funcionario:**

```
INSERT INTO funcionario VALUES (03030030, 03040400, 'F',
'casada', '2017/01/02', 'maria joaquina', 'rua um', 345,
'bairroum', 'campo grande', 7900067, 3345555,
'maria@joaquina.com', 1122323, 'escrevente');
```

```
INSERT INTO funcionario VALUES (03030030, 03040400, 'F',
8888, 'casada', '2017/01/02', 'maria joaquina',
'email@1223.com', 'rua um', 345, 'bairroum', 'campo grande',
7900067, 3345555, '1987/09/09', 1122323, 'escrevente');
```

```
INSERT INTO funcionario VALUES (030677030, 03333400, 'M',
9999, 'casado', '2010/05/07', 'Antonio Joao',
'email@1223.com', 'rua dois', 3454, 'bairrodois', 'campo grande',
7900000, 33355, '1987/09/09', 11299993, 'escrevente');
```

```
INSERT INTO funcionario VALUES (0234030, 054554400, 'F',
93939, 'casada', '2015/01/02', 'maria antonia',
'email@1223.com', 'rua tres', 3445, 'bairrodois', 'campo grande',
7933067, 3345565, '1987/09/09', 1122223, 'escrevente');
```

```
INSERT INTO funcionario VALUES (03045550, 030445320, 'F',
939393, 'solteira', '2017/01/05', 'ana
Pereira', 'email@1223.com', 'rua quatro', 3245, 'bairrodois',
```

```
'campo grande', 7933067, 334444555, '1987/09/09', 134342323,
'escrevente');
```

```
INSERT INTO funcionario VALUES (03343240, 03345340, 'F',
999939,'casada', '2008/01/02', 'maria joaquina silva',
'email@1223.com','rua um', 345, 'bairroum', 'campo grande',
7934567, 3334555, '1987/09/09', 143534523,'auxiliar');
```

```
INSERT INTO funcionario VALUES (0303453430, 03345500, 'M',
838383,'casado', '2010/05/07', 'Antonio Joao silva',
'email@1223.com','rua doiss', 3445, 'bairrodois', 'campo
grande', 79045450, 345455, '1987/09/09', 145454593, 'auxiliar');
```

```
INSERT INTO funcionario VALUES (0545450, 05454500, 'F', 93939,
'casada', '2015/01/02', 'maria antonia rosa','email@1223.com',
'rua tre', 34445, 'bairrodois',' campo grande', 793454545,
3344545, '1987/09/09', 1434343, 'auxiliar');
```

```
INSERT INTO funcionario VALUES (03453454, 030445450, 'F',
83388,'divorciada', '2017/01/05','Ana Pereira de paula',
'email@1223.com','rua quatro', 3245, 'bairrodois', 'campo
grande', 74343467, 33434555, '1987/09/09', 134342323,
'auxiliar');
```

### **Povoando pedido:**

```
INSERT INTO pedidoVALUES ('001', 'certidao', '2017/04/02',
'29839283', '309490300', 'func1');
```

```
INSERT INTO pedido VALUES ('002', 'certidao', '2017/02/02',
'9382009', '3949849', 'func2');
```

```
INSERT INTO pedido VALUES ('003', 'cancelamento',
'2017/01/02', '9929392', '0303030', 'func2');
```

```
INSERT INTO pedido VALUES ('004', 'cancelamento', '2017/02/03',
'390393', '03940933', 'func3');
```

### **Povoando tribunal:**

```
INSERT INTO tribunal VALUES ('001', 'tribunal1', 'cg', 'ms',
'estadual');
```

```
INSERT INTO tribunal VALUES ('002', 'tribunal2', 'brasilia',
'df', 'federal');
```

```
INSERT INTO tribunal VALUES ('003', 'tribunal3 ', 'sao paulo',
'sp', 'federal');
```

```
INSERT INTO tribunalVALUES ('004', 'tribunal4', 'cg', 'ms', 'estadual');
```

#### **Povoando oficio:**

```
INSERT INTO oficio VALUES ('00111', '2017/09/02', '001', '0011', '0011');
```

```
INSERT INTO oficio VALUES ('00222', '2017/09/02', '002', '0022', '0022');
```

```
INSERT INTO oficio VALUES ('00333', '2017/09/02', '003', '0033', '0033');
```

```
INSERT INTO oficioVALUES ('00444', '2017/09/02', '004', '0044', '0044');
```

#### **Povoando oficio\_de\_resposta:**

```
INSERT INTO oficio_de_resposta VALUES ('001', 'text', '2017/09/02', '001');
```

```
INSERT INTO oficio_de_resposta VALUES ('002', 'text', '2017/09/02', '001');
```

```
INSERT INTO oficio_de_resposta VALUES ('003', 'text', '2017/09/02', '002');
```

```
INSERT INTO oficio_de_resposta` VALUES ('004', 'text', '2017/09/02', '003');
```

#### **Povoando instrumento\_de\_protesto:**

```
INSERT INTO instrumento_de_protesto VALUES ('2270-271', 'comum', 'carta AR', '2017/02/03', '3434', '11');
```

```
INSERT INTO instrumento_de_protesto VALUES ('2270-272', 'comum', 'carta AR', '2017/02/03', '45656', '22');
```

```
INSERT INTO instrumento_de_protesto VALUES ('2270-273', 'falimentar', 'edital', '2017/02/03', '45666', '33');
```

```
INSERT INTO instrumento_de_protesto VALUES ('2270-274', 'falimentar', 'edital', '2017/02/03', '555', '44');
```

#### **Povoando processo\_de\_cancelamento:**



```
INSERT INTO processo_de_cancelamento VALUES ('001', '4343',  
'1', '1', '2017/09/03');
```

```
INSERT INTO processo_de_cancelamento VALUES ('002', '554545',  
'1', '1', '2017/09/03');
```

```
INSERT INTO processo_de_cancelamento VALUES ('003', '3434',  
'2', '2', '2017/09/03');
```

```
INSERT INTO processo_de_cancelamento VALUES ('4', '43434', '2',  
'4', '2017/09/03');
```

#### **Povoando certidao\_cancelamento:**

```
INSERT INTO certidao_cancelamento VALUES ('001', '0022',  
'200');
```

```
INSERT INTO certidao_cancelamento VALUES ('002', '0033',  
'300');
```

```
INSERT INTO certidao_cancelamento VALUES ('003', '0044',  
'400');
```

```
INSERT INTO certidao_cancelamento VALUES ('004', '0055',  
'500');
```

#### **Povoando certidao\_protesto:**

```
INSERT INTO certidao_protesto VALUES ('001', '001');
```

```
INSERT INTO certidao_protesto VALUES ('002', '001');
```

```
INSERT INTO certidao_protesto VALUES ('003', '002');
```

```
INSERT INTO certidao_protestoVALUES ('004', '004');
```

#### **Povoando auxiliar:**

```
INSERT INTO auxiliar VALUES ('6499965', 'maria joaquina silva',  
'03345340', 'F', '354345', 'email', 'end', '34234', '10000',  
'2008/01/02', 'solteira', 'funcao1', 'A');
```

```
INSERT INTO auxiliarVALUES ('4545', 'Antonio Joao silva',  
'03345500', 'M', '345345', 'email', 'end', '234', '202002',  
'2010/05/07', 'casado', 'funcao2', 'A');
```

```
INSERT INTO auxiliar VALUES ('78777', 'maria antonia rosa',  
'05454500', 'F', '34534', 'email', 'end', '324234', '9999',  
'2015/01/02', 'casada', 'funcao1', 'B');
```

```
INSERT INTO auxiliar VALUES ('89977', 'Ana Pereira de paula',  
'030445450', 'F', '345435', 'eamil', 'end', '234234', '800',  
'2017/01/05', 'solteira', 'funcao3', 'B');
```

## Consultas

### Consulta 1)

#### Descrição:

Um funcionário deseja saber qual o total de títulos protestados no dia 2017-02-03.

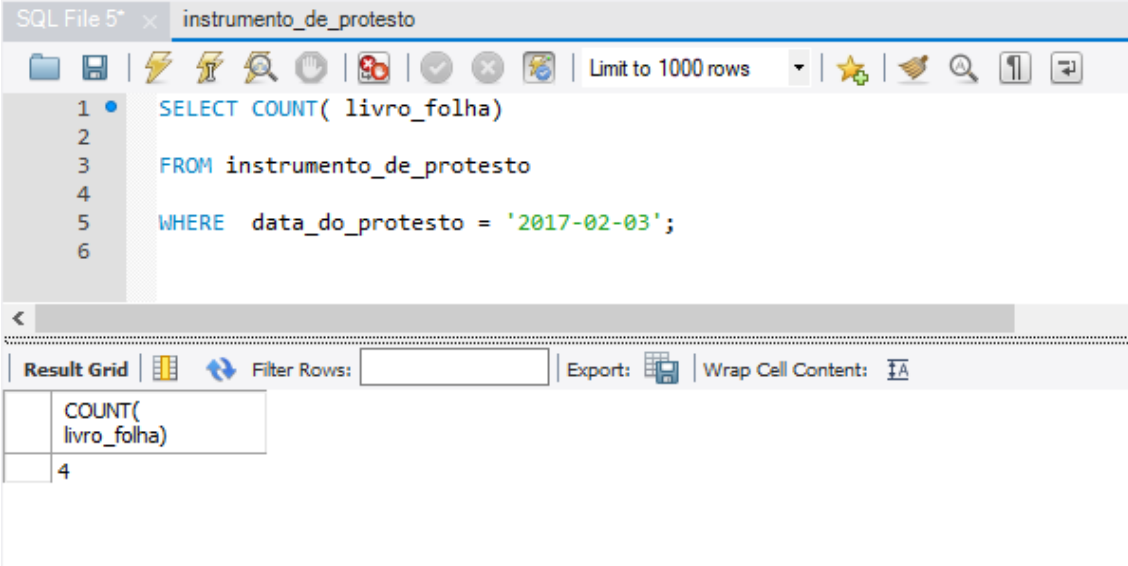
Resultado = 4

#### Implementação:

```
SELECT COUNT( livro_folha)
```

```
FROM instrumento_de_protesto
```

```
WHERE data_do_protesto = '2017-02-03';
```



The screenshot shows a SQL IDE window titled "instrumento\_de\_protesto". The query editor contains the following SQL code:

```
1 SELECT COUNT( livro_folha)
2
3 FROM instrumento_de_protesto
4
5 WHERE data_do_protesto = '2017-02-03';
6
```

Below the query editor, the "Result Grid" is displayed, showing the result of the query:

COUNT( livro_folha)
4

## Consulta 2)

### Descrição:

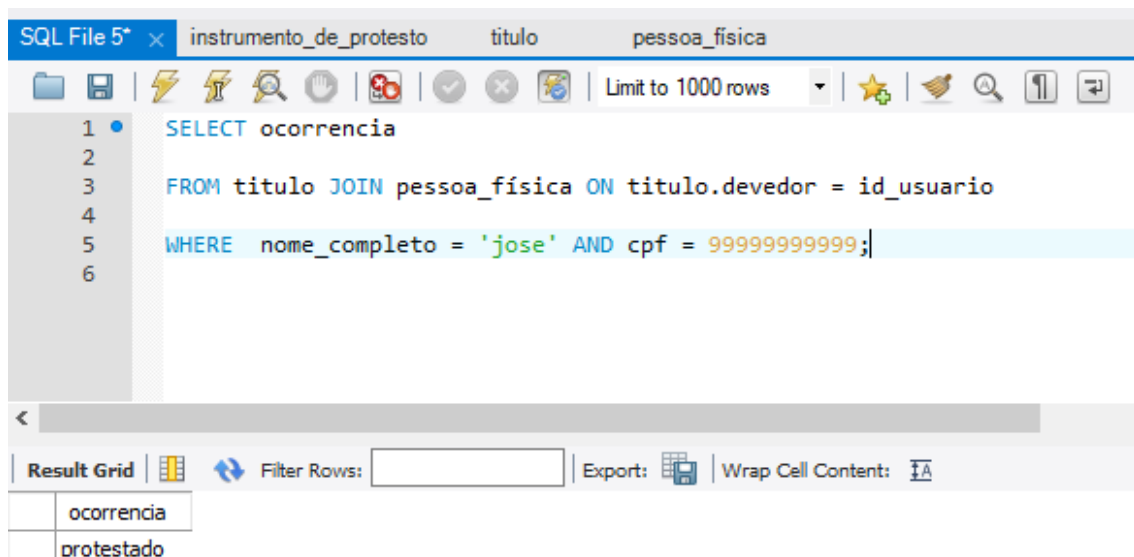
Um usuário (de nome “José Afonso”, com o cpf “99999999999”) deseja saber qual é o estado(a ocorrência) atual de um determinado título em que ele é o devedor. Resposta: Portestado

### Implementação:

```
SELECT ocorrencia
```

```
FROM titulo JOIN pessoa_física ON titulo.devedor = id_usuario
```

```
WHERE nome_completo = 'jose' AND cpf = 99999999999;
```



### Consulta 3)

#### Descrição:

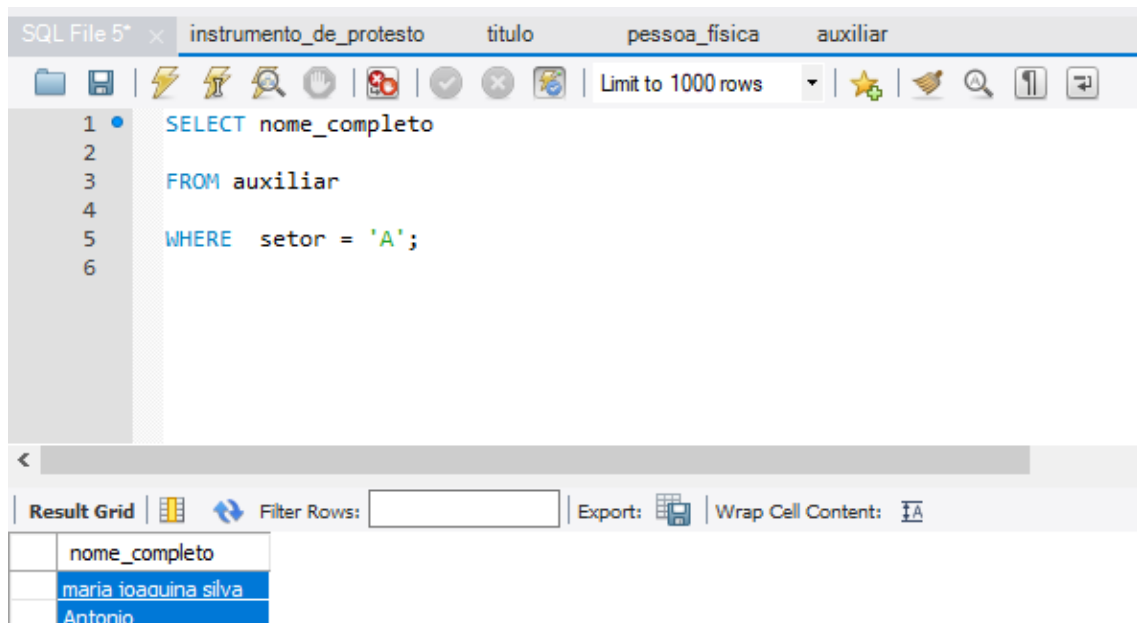
O tabelião deseja saber o nome dos funcionários responsáveis atualmente por operar o setor A. Resposta: maria jaoquina silva e antonio

#### Implementação:

```
SELECT nome_completo
```

```
FROM auxiliar
```

```
WHERE setor = 'A';
```



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 SELECT nome_completo
2
3 FROM auxiliar
4
5 WHERE setor = 'A';
6
```

The result grid below the query editor displays the following data:

nome_completo
maria joaquina silva
Antonio

## Consulta 4)

### Descrição:

O sistema precisa saber quais os títulos protestados existem em nome de um devedor (de nome “banco do brasil”, com cpnj “28928”), para poder gerar uma Certidão de Protesto.

### Implementação:

```
SELECT livro_folha
```

```
FROM
```

```
instrumento_de_protesto AS ip JOIN titulo AS t
```

```
ON ip.protocolo_titulo = t.protocolo_titulo
```

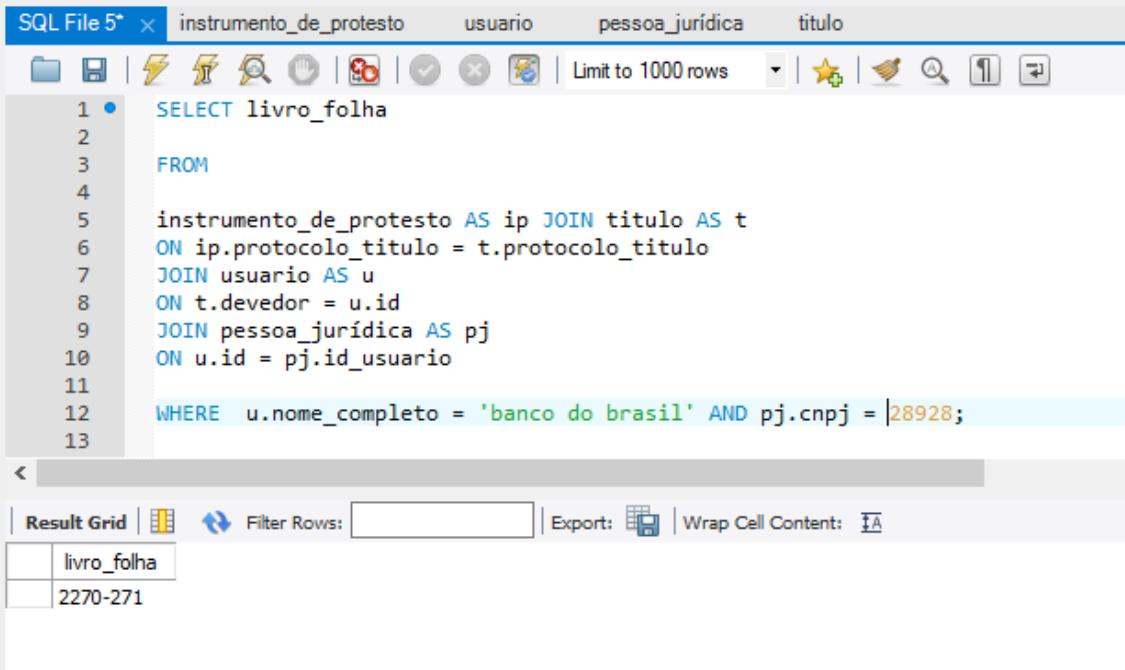
```
JOIN usuario AS u
```

```
ON t.devedor = u.id
```

```
JOIN pessoa_jurídica AS pj
```

```
ON u.id = pj.id_usuario
```

```
WHERE u.nome_completo = 'banco do brasil' AND pj.cnpj = 28928;
```



The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
1 SELECT livro_folha
2
3 FROM
4
5 instrumento_de_protesto AS ip JOIN titulo AS t
6 ON ip.protocolo_titulo = t.protocolo_titulo
7 JOIN usuario AS u
8 ON t.devedor = u.id
9 JOIN pessoa_jurídica AS pj
10 ON u.id = pj.id_usuario
11
12 WHERE u.nome_completo = 'banco do brasil' AND pj.cnpj = 28928;
13
```

The result grid shows the following data:

livro_folha
2270-271

## Consulta 5)

### Descrição:

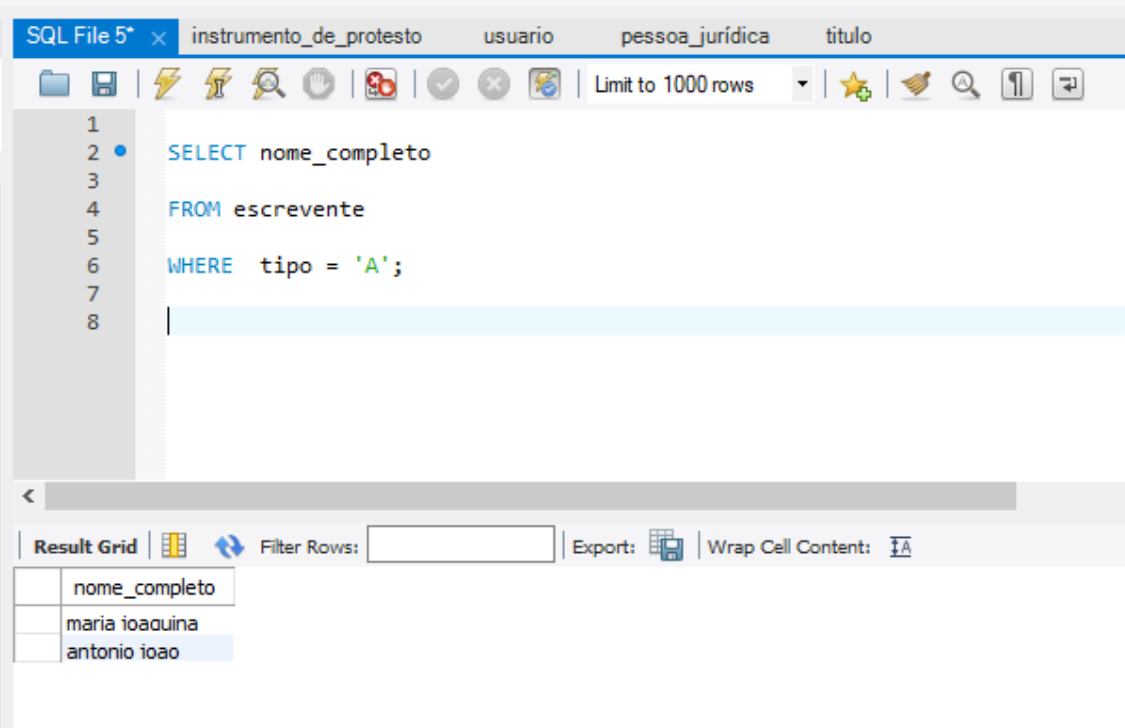
O tabelião deseja saber quem é o Escrevente do tipo "A" atualmente.

### Implementação:

```
SELECT nome_completo
```

```
FROM escrevente
```

```
WHERE tipo = 'A';
```



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1  
2 • SELECT nome_completo  
3  
4 FROM escrevente  
5  
6 WHERE tipo = 'A';  
7  
8
```

The result grid below the query editor displays the following data:

nome_completo
maria ioaquina
antonio ioao

## Consulta 6)

### Descrição:

Tabelião deseja saber qual tribunal enviou o ofício referente ao título que, no dia 21/05/2016, teve como apresentante o portador do rg: 999999999.

### Implementação:

```
SELECT nome
```

```
FROM tribunal  
AS trib JOIN oficio AS o
```

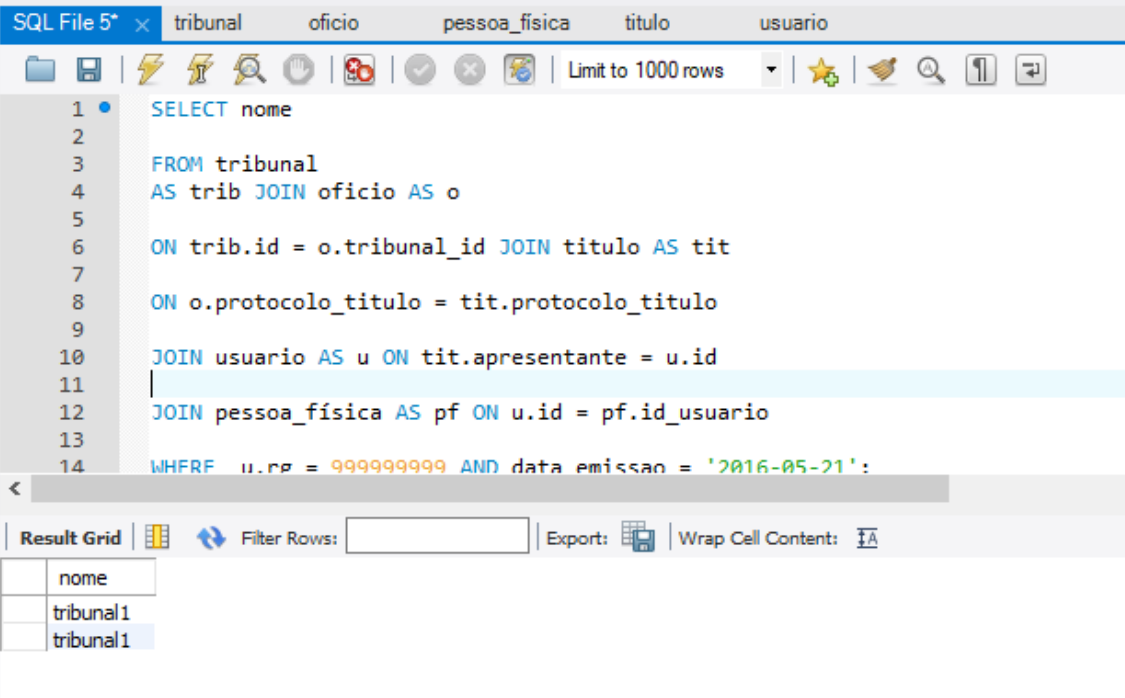
```
ON trib.id = o.tribunal_id JOIN titulo AS tit
```

```
ON o.protocolo_titulo = tit.protocolo_titulo
```

```
JOIN usuario AS u ON tit.apresentante = u.id
```

```
JOIN pessoa_fisica AS pf ON u.id = pf.id_usuario
```

```
WHERE u.rg = 999999999 AND data_emissao = '2016-05-21';
```



The screenshot shows a SQL query editor with the following SQL code:

```
1 SELECT nome  
2  
3 FROM tribunal  
4 AS trib JOIN oficio AS o  
5  
6 ON trib.id = o.tribunal_id JOIN titulo AS tit  
7  
8 ON o.protocolo_titulo = tit.protocolo_titulo  
9  
10 JOIN usuario AS u ON tit.apresentante = u.id  
11  
12 JOIN pessoa_fisica AS pf ON u.id = pf.id_usuario  
13  
14 WHERE u.rg = 999999999 AND data_emissao = '2016-05-21';
```

The editor interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section at the bottom. The "Result Grid" shows the following data:

nome
tribunal1
tribunal1