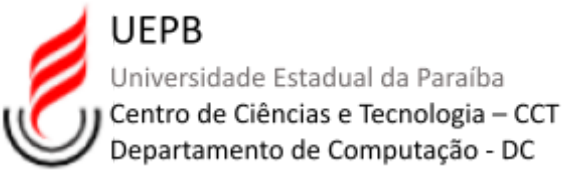


DISCIPLINA: BANCO DE DADOS

Modelo de Entidade Relacionamento - Sistema de controle de candidaturas e gastos das legislaturas

(2022)

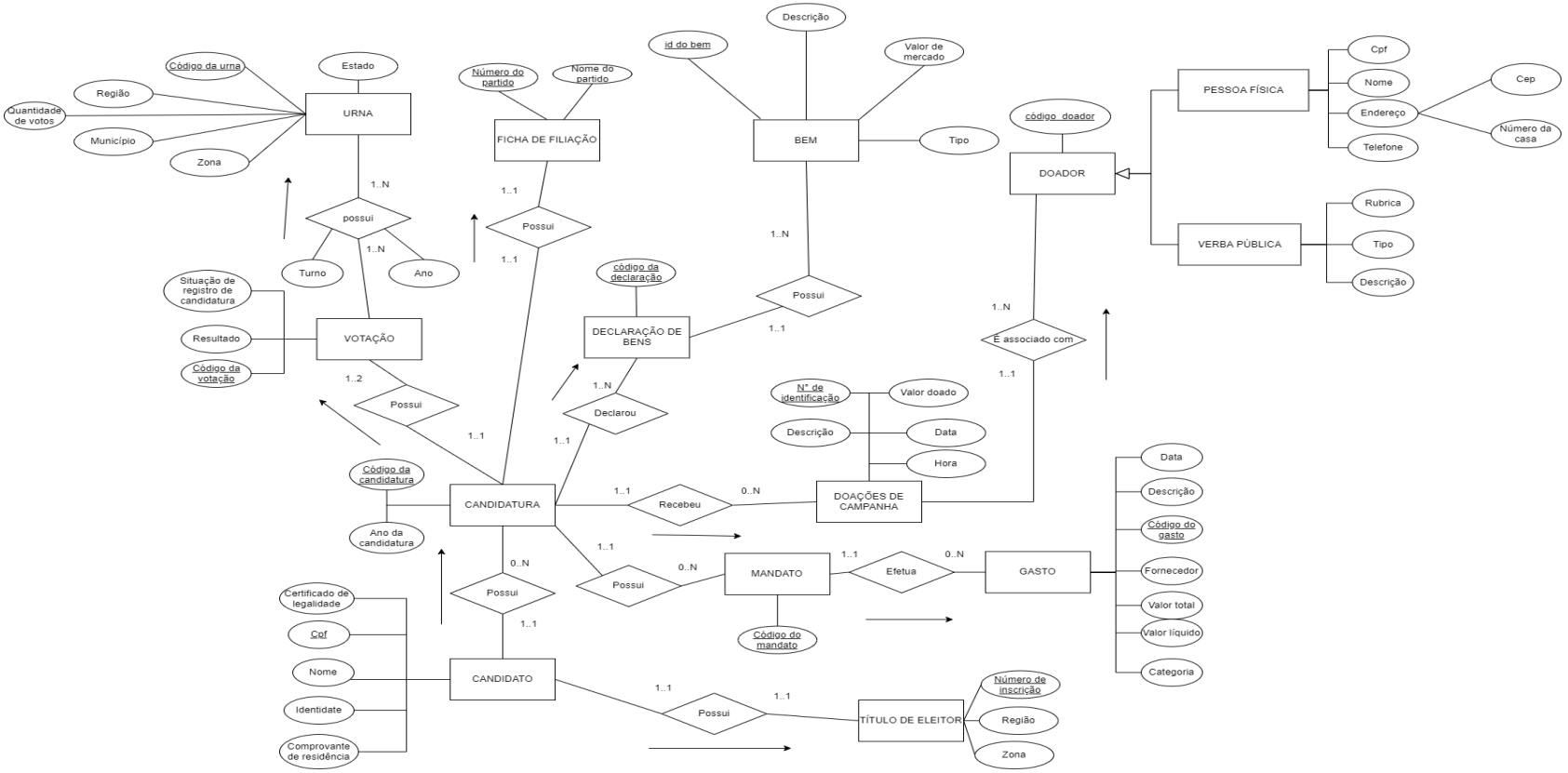
ALUNOS:

- **Jefferson Gomes de Almeida - 192080016;**
- **Ângelo Gabriel Paz da Silva - 192080075;**
- **Kennedy Johnson de Sousa Dantas - 192080156**

Prof(ª). Fabio Leite

CAMPUS CAMPINA GRANDE, Março de 2022

Diagrama de entidade e relacionamento



Projeto relacional

Candidato(cpf_candidato, nome_candidato, identidade_candidato, comprovante_de_residência_candidato, certificado_de_legalidade_candidato, #número_de_inscição_título_de_eleitor*)

Título_de_eleitor(número de inscrição_título_de_eleitor, zona_de_inscrição_titulo_de_eleitor, região_titulo_de_eleitor)

Candidatura(código_da_candidatura, ano_da_candidatura,#número_do_partido_ficha_de_filiação*,#cpf candidato*)

Mandato(código_do_mandato, #código_da_candidatura*)

Ficha_de_filiação(número_do_partido_ficha_de_filiação, nome_do_partido_ficha_de_filiação)

Declaração_de_bens(código_da_declaração, #código_da_candidatura*)

Bem(id_do_bem, tipo_do_bem, descrição_do_bem, valor_de_mercado_do_bem, #código_da_declaração_de_bens*)

Votação(código_da_votação, resultado_da_votação, situação_de_registro_de_candidatura_da_votação, #código_da_candidatura*)

Votação_possui_urna(#código_da_votação,#código_da_urna,turno,ano)

Urna(código_da_urna, quantidade_de_votos_da_urna, zona_da_urna, região_da_urna, município_da_urna, estado_da_urna)

Gasto(código_do_gasto,data_gasto, descrição_do_gasto, fornecedor_do_gasto, valor_total_do_gasto, valor_liquido_gasto, categoria_do_gasto ,#código_do_mandato*)

Doações_de_campanha(num_de_identificação_doação ,valor_doado_doação,descrição_doação, data_doação, hora_doação,#código_da_candidatura*)

Doador(código_doador, #num_de_identificação_doação*)

Pessoa_física(#código_doador,cpf_pessoa_física, nome_pessoa_física, Cep_pessoa_física, Numero_residencia_pessoa_física, telefone_pessoa_física)

Verba_pública(#código_doador, rúbrica_verba_pública, tipo_verba_pública, descrição_verba_pública)

Dicionário de dados

Entidade: Candidato

Atributo	Classe	Domínio	Tamanho	Descrição
cpf_candidato	Determinante	Texto	11	Registro candidato, único
nome_candidato	Simples	Texto	50	
identidade_candidato	Simples	Numérico		
comprovante_de_residência_candidato	Simples	Blob	100 kb	Imagem do comprovante de residência do candidato
certificado_de_legalidade_candidato	Simples	Blob	100 kb	Imagem do certificado de legalidade
número_de_inscrição_título_de_eleitor	Simples	Numérico		Chave estrangeira de título de eleitor, obrigatório

Entidade: Título_de_eleitor

Atributo	Classe	Domínio	Tamanho	Descrição
número_de_inscrição_título_de_eleitor	Determinante	Numérico		Registro do número de inscrição do título de eleitor, único
zona_de_inscrição_titulo_de_eleitor	Simples	Numérico		
região_titulo_de_eleitor	Simples	Texto	50	

Entidade: Candidatura

Atributo	Classe	Domínio	Tamanho	Descrição
código_da_candidatura	Determinante	Numérico		Registro do código candidatura, único
ano_da_candidatura	Simples	Numérico		Ano da candidatura
número_do_partido_ficha_de_filiação	Simples	Número		Chave estrangeira de ficha de filiação, atributo obrigatório. Não pode se repetir caso já tenha sido cadastrada em outro pleito.
cpf_candidato	Simples	Texto	11	Chave estrangeira de Candidato, obrigatório

Entidade: Mandato

Atributo	Classe	Domínio	Tamanho	Descrição
código_do_mandato	Determinante	Numérico		Registro do código do mandato, único
código_da_candidatura	Simples	Numérico		Chave estrangeira de candidatura, único, obrigatório

Entidade: Ficha_de_filiação

Atributo	Classe	Domínio	Tamanho	Descrição
número_do_partido_ficha_de_filiação	Determinante	Numérico		Registro da ficha de filiação, único
nome_do_partido_ficha_de_filiação	Simples	Texto	50	

Entidade: Declaração_de_bens

Atributo	Classe	Domínio	Tamanho	Descrição
código_da_declaração	Determinante	Numérico		Registro da declaração, único
código_da_candidatura	Simples	Numérico		Chave estrangeira de candidatura, obrigatório

Entidade: Bem

Atributo	Classe	Domínio	Tamanho	Descrição
id_do_bem	Determinante	Numérico		Registro do bem, único
tipo_do_bem	Simples	Texto	50	Terreno, casa, apartamento, outros imóveis, veículos, licença e concessão especial, quotas e quinhões de capital, joias, obras de arte, ouro ativo financeiro, etc.
descrição_do_bem	Simples	Texto	800	Descrição de como o bem em questão se encontra, este é um atributo opcional, podendo ficar em branco
valor_de_mercado_do_bem	Simples	Numérico		Preço do bem
código_da_declaração_de_bens	Simples	Numérico		Chave estrangeira de declaração de bens, obrigatório

Entidade: Votação

Atributo	Classe	Domínio	Tamanho	Descrição
código_da_votação	Determinante	Numérico		Registro da votação, único
resultado_da_votação	Simples	Texto	50	Eleito, eleito por média, eleito por quociente partidário, indeferido com recurso, não eleito, suplente, substituído, renúncia ou falecimento, registro negado antes da eleição, registro negado depois da eleição
situação_de_registro_de_candidatura_da_votação	Simples	Texto	10	Apto, inapto ou cadastrado
código_da_candidatura	Simples	Numérico		Chave estrangeira de candidatura, único, obrigatório

Entidade: Votação_possui_urna

Atributo	Classe	Domínio	Tamanho	Descrição
código_da_votação	Determinante	Numérico		Chave primária estrangeira, vinda de votação, único
código_da_urna	Determinante	Numérico		Chave primária estrangeira, vinda da urna, único
turno	Simples	Texto	8	deve-se informar como: n° turno
ano	Simples	Numérico		

Entidade: Urna

Atributo	Classe	Domínio	Tamanho	Descrição
código_da_urna	Determinante	Numérico		Registro da urna, único
quantidade_de_votos_da_urna	Simples	Numérico		
zona_da_urna	Simples	Numérico		
região_da_urna	Simples	Texto	50	
município_da_urna	Simples	Texto	50	
estado_da_urna	Simples	Texto	50	

Entidade: Gasto

Atributo	Classe	Domínio	Tamanho	Descrição
código_do_gasto	Determinante	Numérico		Registro do gastro, único
data_gasto	Simples	Data		Formato: aaaa-mm-dd
descrição_do_gasto	Simples	Texto	800	
fornecedor_do_gasto	Simples	Texto	200	
valor_total_do_gasto	Simples	Numérico		
categoria_do_gasto	Simples	Texto	100	
valor_liquido_gasto	Simples	Numérico		
código_do_mandato	Simples	Numérico		Chave estrangeira de mandato, obrigatório

Entidade: Doação De Campanha

Atributo	Classe	Domínio	Tamanho	Descrição
num_de_identificação_doação	Determinante	Numérico		Registro da doação de campanha, único
valor_doado_doação	Simples	Numérico		
descrição_doação	Simples	Texto	800	
data_doação	Simples	Data		Formato: aaaa-mm-dd
hora_doação	Simples	Texto	14	Formato: ## : ## AM/PM
código_da_candidatura	Simples	Numérico		Chave estrangeira de candidatura, obrigatório

Entidade: Doador

Atributo	Classe	Domínio	Tamanho	Descrição
código_doador	Determinante	Numérico		Registro do doador, único
num_de_identificação_doação	Simples	Numérico		Chave estrangeira, doação de campanha, único, obrigatório

Entidade: Pessoa_física

Atributo	Classe	Domínio	Tamanho	Descrição
código_doador	Determinante	Numérico		Chave estrangeira de doador, único
cpf_pessoa_física	Simples	Texto	11	Único, obrigatório
nome_pessoa_física	Simples	Texto	50	
cep_pessoa_físicacen	Simples	Numérico		
número_residência_pessoa_física	Simples	Numérico		
telefone_pessoa_física	Simples	Texto	15	

Entidade: Verba_pública

Atributo	Classe	Domínio	Tamanho	Descrição
código_doador	Determinante	Numérico		Chave estrangeira de doador, único
rúbrica_verba_pública	Simples	Numérico		Código referenciando a origem do recurso, único, obrigatório
tipo_verba_pública	Simples	Texto	50	Repasses oriundos do fundo partidário, doações do próprio partido, etc.
descrição_verba_pública	Simples	Texto	800	

Scripts de criação sql

```
drop database if EXISTS projetobd;
CREATE DATABASE projetobd
DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;
USE projetobd;



CREATE TABLE IF NOT EXISTS Candidato(
	cpf_candidato VARCHAR (11) UNIQUE NOT NULL,
	nome_candidato VARCHAR (50),
	identidade_candidato INTEGER,
	`comprovante_de_residência_candidato` BLOB,
   certificado_de_legalidade_candidato BLOB,
   `número_de_inscrição_título_de_eleitor` INTEGER NOT NULL,
	CONSTRAINT PK_CPF_CANDIDATO PRIMARY KEY (cpf_candidato),
	CONSTRAINT FK_NUM_INS_T_E FOREIGN KEY (`número_de_inscrição_título_de_eleitor`)
REFERENCES `Título_de_eleitor`(`número_de_inscrição_título_de_eleitor`)
);

CREATE TABLE IF NOT EXISTS `Título_de_eleitor`(
	`número_de_inscrição_título_de_eleitor` INTEGER NOT NULL,
	`zona_de_inscrição_titulo_de_eleitor` INTEGER,
	`região_titulo_de_eleitor` VARCHAR (50),
	CONSTRAINT PK_NUM_INSC_T_E PRIMARY KEY
(`número_de_inscrição_título_de_eleitor`)
);

CREATE TABLE IF NOT EXISTS Candidatura(
	`código_da_candidatura` INTEGER NOT NULL AUTO_INCREMENT,
	ano_da_candidatura INTEGER,
	`número_do_partido_ficha_de_filiação` INTEGER NOT NULL,
   cpf_candidato VARCHAR (11) NOT NULL,
   CONSTRAINT UN_ANO_NUMPART UNIQUE(ano_da_candidatura,
`número_do_partido_ficha_de_filiação`),
	CONSTRAINT PK_COD_CAND PRIMARY KEY (`código_da_candidatura`),
   CONSTRAINT FK_NUM_PART_FCH_FIL FOREIGN KEY (`número_do_partido_ficha_de_filiação`)
REFERENCES `Ficha_de_filiação`(`número_do_partido_ficha_de_filiação`),
   CONSTRAINT FK_CPF_CANDIDATO FOREIGN KEY (`cpf_candidato`) REFERENCES
Candidato(`cpf_candidato`)
);



CREATE TABLE IF NOT EXISTS Mandato(
	`código_do_mandato` INTEGER NOT NULL AUTO_INCREMENT,
	`código_da_candidatura` INTEGER NOT NULL UNIQUE,
```

```
        CONSTRAINT PK_COD_MAND PRIMARY KEY (`código_do_mandato`),
   CONSTRAINT FK_COD_CAND FOREIGN KEY (`código_da_candidatura`) REFERENCES
Candidatura(`código_da_candidatura`)
);

CREATE TABLE IF NOT EXISTS `Ficha_de_filiação`(
        `número_do_partido_ficha_de_filiação` INTEGER NOT NULL,
        `nome_do_partido_ficha_de_filiação` VARCHAR (50),
        CONSTRAINT PK_NUM_PART_FCH_FIL PRIMARY KEY
(`número_do_partido_ficha_de_filiação`)
);

CREATE TABLE IF NOT EXISTS `Declaração_de_bens`(
        `código_da_declaração` INTEGER NOT NULL AUTO_INCREMENT,
        `código_da_candidatura` INTEGER NOT NULL,
        CONSTRAINT PK_COD_DCL PRIMARY KEY (`código_da_declaração`),
   CONSTRAINT FK_COD_CAND FOREIGN KEY (`código_da_candidatura`) REFERENCES
Candidatura(`código_da_candidatura`)
);

CREATE TABLE IF NOT EXISTS Bem(
        id_do_bem INTEGER NOT NULL AUTO_INCREMENT,
        tipo_do_bem VARCHAR (50),
        `descrição_do_bem` VARCHAR (800),
   valor_de_mercado_do_bem DOUBLE PRECISION,
   `código_da_declaração_de_bens` INTEGER NOT NULL,
        CONSTRAINT PK_ID_BEM PRIMARY KEY (`id_do_bem`),
        CONSTRAINT FK_COD_DCL_BENS FOREIGN KEY (`código_da_declaração_de_bens`)
REFERENCES `Declaração_de_bens`(`código_da_declaração`)
);

CREATE TABLE IF NOT EXISTS `Votação`(
        `código_da_votação` INTEGER NOT NULL AUTO_INCREMENT,
        `resultado_da_votação` enum('eleito', 'eleito por média', 'eleito por quociente partidário',
'indeferido com recurso', 'não eleito', 'suplente', 'substituído', 'renúncia ou falecimento', 'registro
negado antes da eleição', 'registro negado depois da eleição'),
        `situação_de_registro_de_candidatura_da_votação` enum('apto','inapto','cadastrado'),
   `código_da_candidatura` INTEGER NOT NULL UNIQUE,
        CONSTRAINT PK_COD_VOT PRIMARY KEY (`código_da_votação`),
        CONSTRAINT FK_COD_CAND FOREIGN KEY (`código_da_candidatura`) REFERENCES
Candidatura(`código_da_candidatura`)
);

CREATE TABLE IF NOT EXISTS `Votação_possui_urna`(
        `código_da_votação` INTEGER NOT NULL,
        `código_da_urna` INTEGER NOT NULL,
        turno VARCHAR (8),
   ano INTEGER,

   CONSTRAINT FK_COD_VOT FOREIGN KEY (`código_da_votação`) REFERENCES
`Votação`(`código_da_votação`),
```

```
   CONSTRAINT FK_COD_URNA FOREIGN KEY (`código_da_urna`) REFERENCES
`Urna`(`código_da_urna`),
   CONSTRAINT PK_COD_VOT PRIMARY KEY (`código_da_votação`, `código_da_urna`)

);

CREATE TABLE IF NOT EXISTS Urna(
        `código_da_urna` INTEGER NOT NULL AUTO_INCREMENT,
        `quantidade_de_votos_da_urna` INTEGER,
        zona_da_urna INTEGER,
   `região_da_urna` VARCHAR (50),
   `município_da_urna` VARCHAR (50),
   estado_da_urna VARCHAR (50),
   CONSTRAINT PK_COD_URNA PRIMARY KEY (`código_da_urna`)
);

CREATE TABLE IF NOT EXISTS Gasto(
        `código_do_gasto` INTEGER NOT NULL AUTO_INCREMENT,
        data_gasto DATE,
        `descrição_do_gasto` VARCHAR (800),
   fornecedor_do_gasto VARCHAR (200),
   valor_total_do_gasto DOUBLE PRECISION,
   categoria_do_gasto  VARCHAR (100),
   valor_liquido_gasto DOUBLE PRECISION,
   `código_do_mandato` INTEGER NOT NULL,
   CONSTRAINT PK_COD_GAST PRIMARY KEY (`código_do_gasto`),
        CONSTRAINT FK_COD_MDT FOREIGN KEY (`código_do_mandato`) REFERENCES
`Mandato`(`código_do_mandato`)
);

CREATE TABLE IF NOT EXISTS `Doação_de_campanha`(
        `num_de_identificação_doação` INTEGER NOT NULL AUTO_INCREMENT,
        `valor_doado_doação` DOUBLE PRECISION,
        `descrição_doação` VARCHAR (800),
   `data_doação`DATE,
   `hora_doação`VARCHAR (14),
   `código_da_candidatura`  INTEGER NOT NULL,
   CONSTRAINT PK_NUM_IDT_DOA PRIMARY KEY (`num_de_identificação_doação`),
        CONSTRAINT FK_COD_CAND FOREIGN KEY (`código_da_candidatura`) REFERENCES
`Candidatura`(`código_da_candidatura`)
);

CREATE TABLE IF NOT EXISTS Doador(
        `código_doador` INTEGER NOT NULL AUTO_INCREMENT,
        `num_de_identificação_doação` INTEGER NOT NULL,
        CONSTRAINT PK_COD_DOA PRIMARY KEY (`código_doador`),
        CONSTRAINT FK_NUM_IDT_DOA FOREIGN KEY (`num_de_identificação_doação`)
REFERENCES Doação_de_campanha(`num_de_identificação_doação`)
);

CREATE TABLE IF NOT EXISTS `Pessoa_física`(
        `código_doador` INTEGER NOT NULL,
```

```
        `cpf_pessoa_física` VARCHAR (11) UNIQUE,
        `nome_pessoa_física` VARCHAR (50),
        `cep_pessoa_física` INTEGER,
   `número_residência_pessoa_física` INTEGER,
   `telefone_pessoa_física` VARCHAR (15),

   CONSTRAINT PK_COD_DOA PRIMARY KEY (`código_doador`),
        CONSTRAINT FK_COD_DOA FOREIGN KEY (`código_doador`) REFERENCES
Doador(`código_doador`)
);

CREATE TABLE IF NOT EXISTS `Verba_pública`(
        `código_doador` INTEGER NOT NULL,
        `rúbrica_verba_pública` INTEGER,
        `tipo_verba_pública` VARCHAR (50),
        `descrição_verba_pública` VARCHAR (800),
   CONSTRAINT PK_COD_DOA PRIMARY KEY (`código_doador`),
        CONSTRAINT FK_COD_DOA FOREIGN KEY (`código_doador`) REFERENCES
Doador(`código_doador`)
);
```

Scripts de Inserção sql

```
USE projetobd;


INSERT INTO Candidato(cpf_candidato, nome_candidato ,identidade_candidato,
número_de_inscrição_título_de_eleitor)
VALUES
(2345678910, 'João', 13141341, 1111),
(2345678911, 'Maicon', 13141342, 2222),
(2345678912, 'Jonas', 13141343, 3333),
(2345678913, 'Horacio', 13141344, 4444),
(2345678914, 'Marcia', 13141345, 5555),
(2345678915,'Leticia',13141346,6666),
(12345678916,'Thiago', 13141347,      7777),
(22345678917,'Paulo',   13141348,      8888),
(22345678918,'Renato', 13141349,      9999),
(22345678919,'Marcos', 13141350,      1010),
(22345678920,'Bruno',   13141351,      11111);

INSERT INTO Título_de_eleitor(número_de_inscrição_título_de_eleitor,
zona_de_inscrição_titulo_de_eleitor , região_titulo_de_eleitor)
VALUES
(1111, 1, 'Norte'),
(2222, 2, 'Nordeste'),
(3333, 3, 'Nordeste'),
(4444, 4, 'Nordeste'),
(5555, 6, 'Nordeste'),
```

```
(6666,   7,        'Nordeste'),
(7777,   6,        'Nordeste'),
(8888,   8,        'Norte'),
(9999,   9,        'Norte'),
(1010,   10,       'Norte'),
(11111,  11,       'Norte');

INSERT INTO Candidatura(ano_da_candidatura,
número_do_partido_ficha_de_filiação,cpf_candidato)
VALUES
('2018',1,'2345678910'),
('2018',2,'2345678911'),
('2018',3,'2345678912'),
('2018',4,'2345678913'),
('2018',5,'2345678914'),
('2018',6,'2345678915'),
('2018',7,'12345678916'),
('2022',8,'22345678917'),
('2022',9,'22345678918'),
('2018',10,'22345678919'),
('2022',11,'22345678920');

INSERT INTO Mandato(código_da_candidatura)
VALUES
(0001),
(0002),
(0003),
(0004),
(0005),
(0006),
(0007),
(0008),
(0009),
(0010),
(0011);

INSERT INTO Ficha_de_filiação(número_do_partido_ficha_de_filiação,
nome_do_partido_ficha_de_filiação)
VALUES
(1 ,  'CDU'),
(2 ,  'Partido de direita'),
(3 , 'Partido de esquerda'),
(4 ,'TARD'),
(5 , 'Partido central'),
(6,'BTA'),
(7,        'WOW'),
(8,        'JOOJ'),
(9,        'PNG'),
(10,'CDU'),
(11,'CDU');

INSERT INTO Declaração_de_bens(`código_da_candidatura`)
```

```
VALUES
(1),
(2),
(3),
(4),
(5),
(6),
(7),
(8),
(9),
(10),
(11);

INSERT INTO Bem(tipo_do_bem, descrição_do_bem,
valor_de_mercado_do_bem,código_da_declaração_de_bens)
VALUES
('Imóvel',    'Casarão Branco',  '800000.00',    1),
('Automóvel',    'BMW 2015',    '290000.00',    2),
('Imóvel',    'Casarão Cinza',    '80000.00',    3),
('Joias',    'Bonitas',    '12000.00', 4),
('Ouro ativo',    'Brilhante',    '10000.00', 5),
('Imóvel', 'top', 120000.00,5),
('Imóvel',          'top',     120000.00,       6),
('Imóvel',          'Apartamento',   400000.00,       7),
('Ouro ativo',     'Ouro',   10000.00,         8),
('Ouro ativo',     'Ouro',   36000.00,         9),
('Imóvel',          'Apartamento',   512000.00,       10),
('Automóvel',     'Fusca', 44000.00,         11);

INSERT INTO `Votação`(`resultado_da_votação`,
`situação_de_registro_de_candidatura_da_votação`, `código_da_candidatura`)
VALUES
('eleito', 'apto' , 1),
('eleito', 'apto' , 2),
('suplente','apto', 3),
('eleito', 'apto' , 4),
('eleito',  'apto', 5),
('eleito',  'apto', 6),
('eleito', 'apto',    7),
('eleito', 'apto',    8),
('eleito', 'apto',    9),
('eleito', 'apto',    10),
('eleito', 'apto',    11);

INSERT INTO `Votação_possui_urna`(`código_da_votação`, `código_da_urna`,`turno`,`ano`)
VALUES
(1,1,   '1°',    '2018'),
(2,2,    '1°',    '2018'),
(3,3,    '1°',    '2018'),
(4,4,    '2°',    '2018'),
(5,5,   '1°',   '2018'),
(6,6,   '1°',   '2018'),
```

```
(6,7,   '1°',   '2018'),
(7,     7,      '2°',   2018),
(8,     8,      '1°',   2018),
(9,     9,      '2°',   2018),
(10,12, '2°',   2018),
(11,13, '1°',   2018);

INSERT INTO Urna(`quantidade_de_votos_da_urna`,zona_da_urna,`região_da_urna`
,`município_da_urna` ,estado_da_urna)
VALUES
('50000',   1 ,  'Nordeste',    'Campina Grande',    'Paraíba'),
('60000',   2,   'Nordeste' ,   'Campina Grande',    'Paraíba'),
('60000',   3,   'Nordeste',    'Campina Grande',   'Paraíba'),
('80000',   4,   'Oeste',    'Campina Grande',   'Paraíba'),
('40000',  5,    'Sul',    'Campina Grande',    'Paraíba'),
('500',   6,    'Sul',    'Campina Grande',    'Paraíba'),
('10',    6,    'Norte',    'Campina Grande',    'Paraíba'),
('500',   6,        'Oeste', 'Campina Grande',    'Paraíba'),
('800',   7,        'Leste',  'Campina Grande',    'Paraíba'),
('8000',8,          'Nordeste',         'Campina Grande',    'Paraíba'),
('10000',9,         'Nordeste',         'Campina Grande',    'Paraíba'),
('70000',10,'Sul',            'Queimadas',    'Paraíba'),
('90000',11,'Sul','Barra de Santana',    'Paraíba');

INSERT INTO Gasto(data_gasto,`descrição_do_gasto`,fornecedor_do_gasto,valor_total_do_gasto
,categoria_do_gasto ,valor_liquido_gasto ,`código_do_mandato`)
VALUES
('2018-03-10','Carreata','Fundo eleitoral',14000000.00,'propaganda',1400000.00,1),
('2018-03-11','Propaganda eleitoral','Fundo eleitoral',13000000.00,'propaganda',1300000.00,2),
('2018-03-12','Carreata','Fundo eleitoral',12000000.00,'propaganda',1200000.00,3),
('2018-03-13','Carreata','Fundo eleitoral',11000000.00,'propaganda',1100000.00,4),
('2018-03-14','Carreata','Fundo eleitoral',10000000.00,'propaganda',1000000.00,5),
('2018-03-15','Carreata','Fundo eleitoral',500.00,'propaganda',200.00,6),
('2018-03-15','Carreata','Fundo eleitoral',500.00,'propaganda',100.00,5),
('2018-03-15',    'Outdoors',         'Fundo eleitoral',            9000000.00,     'propaganda',    6000.00,
        6),
('2018-03-16',    'Propaganda eleitoral',    'Fundo eleitoral',            8000000.00,     'propaganda',
6000.00,7),
('2018-03-17',    'Outdoors',         'Fundo eleitoral',            7000000.00,     'propaganda',    2000.00,
        8),
('2018-03-18',    'Outdoors',         'Fundo eleitoral',            5000000.00,     'propaganda',    1000.00,
        9),
('2018-03-19',    'Propaganda eleitoral',    'Fundo eleitoral',            4000000.00,     'propaganda',
6000.00,        10),
('2018-03-20',    'Propaganda eleitoral',    'Fundo eleitoral',            3000000.00,     'propaganda',
1000.00,        11);

INSERT INTO
`Doação_de_campanha`(`valor_doado_doação`,`descrição_doação`,`data_doação`,`hora_doação`,`c
ódigo_da_candidatura`)
VALUES
(50000.00,    'doação para carreata',    '2018-03-01',    '18:40 PM',1),
```

```
(40000.00,    'doação para propaganda',    '2018-03-01',    '20:00 PM',2),
(30000.00,    'doação para outdoors',    '2018-03-01',    '09:35 AM',3),
(20000.00,    'doação de bezerro',    '2018-03-01',    '06:00 AM',4),
(10000.00,    'doação para carreata',    '2018-03-01',    '20:00 PM',5),
(10000.00,        'doação para propaganda',        '2018-03-01',    '20:00 PM',        6),
(10000.00,        'doação para outdoors',    '2018-03-01',    '18:00 PM',        7),
(10000.00,        'doação para propaganda',        '2018-03-01',    '08:00 AM',        8),
(10000.00,        'doação para propaganda',        '2018-03-01',    '09:35 AM',        9),
(10000.00,        'doação para outdoors',    '2018-03-01',    '09:35 AM',        10),
(10000.00,        'doação para outdoors',    '2018-03-01',    '12:30 AM',        11);

INSERT INTO Doador(`num_de_identificação_doação`)
VALUES
(1),
(2),
(3),
(4),
(5),
(6),
(7),
(8),
(9),
(10),
(11);

INSERT INTO
`Pessoa_física`(`código_doador`,`cpf_pessoa_física`,`nome_pessoa_física`,`cep_pessoa_física`,`nú
mero_residência_pessoa_física`,`telefone_pessoa_física`)
VALUES
(0001,'11389128016','Roberto Pereira',19314416,08,'83988449832'),
(0002,'07745718023','Antonio da Silva',17815360,04,'83988789865'),
(0003,'11191332063','Nathan Orestes',18063520,140,'839886098476'),
(0004,'54789321125','Leonardo Alves',18961010,54,'839994512782'),
(0006,'11111111116','Ash Solsa',    111111116,        6,        '83911111116'),
(0007,'11111111117',        'Joseph Joestar',        111111117,        7,        '83911111117'),
(0008,'11111111118',        'Ana Banana',    111111118,        8,        '83911111118'),
(0009,'11111111119',        'Vitor Romano',    111111119,        9,        '83911111119'),
(0010,'11111111110',        'Rebeca Riba',    111111110,10,    '83911111110'),
(0011,'11111111102',        'Isabelly Gomes',        111111102,        11,        '84911111111');

INSERT INTO `Verba_pública`(`código_doador`,`rúbrica_verba_pública` ,`tipo_verba_pública`
,`descrição_verba_pública` )
VALUES
(0001,2345,'repasses oriundos do fundo partidário','auxílio para candidatura'),
(0002,5678,'doações do próprio partido','auxílio para candidatura'),
(0003,4356,'repasses oriundos do fundo partidário','auxílio para candidatura'),
(0004,1236,'doações do próprio partido','auxílio para candidatura'),
(0005,    500,        'doações do próprio partido',        'auxílio para candidatura'),
(0006,    600,        'doações do próprio partido',        'auxílio para candidatura'),
(0007,    700,        'doações do próprio partido',        'auxílio para candidatura'),
(0008,    800,        'doações do próprio partido',        'auxílio para candidatura'),
(0009,    900,        'doações do próprio partido',        'auxílio para candidatura'),
```

```
(0010,   1000,    'doações do próprio partido',      'auxílio para candidatura'),
(0011,   1100,    'doações do próprio partido',      'auxílio para candidatura');
(0011,   1100,    'doações do próprio partido',      'auxílio para candidatura');
```

Scripts de consulta sql

```
USE projetobd;

-- A)
-- Liste os candidatos pelo nome e CPF que obtiveram votos na cidade de "Campina Grande" e que gastaram mais de R$ 5.000,00 na campanha;

SELECT c.nome_candidato as 'Nome', c.cpf_candidato as 'CPF'
from gasto as g, candidato as c,mandato as m,candidatura as ctra, votação as v, urna as u, votação_possui_urna as vu
where u.município_da_urna = 'Campina Grande' and u.código_da_urna = vu.código_da_urna
and vu.código_da_votação = v.código_da_votação and v.código_da_candidatura = m.código_da_candidatura
and m.código_do_mandato = g.código_do_mandato and g.valor_total_do_gasto and ctra.código_da_candidatura = v.código_da_candidatura and ctra.cpf_candidato = c.cpf_candidato
group by  c.nome_candidato;

-- B)
-- Liste os candidatos pelo nome e CPF e código do partido, total declarado, total recebido em doação e total gasto em campanha que possuíram mais de 500 votos na eleição de 2018 na cidade de Campina Grande;

SELECT
	TEMP.cpf_candidato,
	TEMP.nome_candidato,
	TEMP.código_da_candidatura,
	TEMP.TOTAL AS TOTAL_DECLARADO,
	TEMP.QTD_VOTOS,
	SUM(DC.valor_doado_doação) AS VALOR_TOTAL_DOACAO,
	SUM(G.valor_total_do_gasto) AS VALOR_TOTAL_DO_GASTO
FROM (
	SELECT
	-- *
		TEMP1.cpf_candidato,
		TEMP1.nome_candidato,
		TEMP1.código_da_candidatura,
		SUM(TEMP1.QTD_VOTOS) AS QTD_VOTOS,
		SUM(TEMP1.SOMA) AS TOTAL
	FROM (
		SELECT
				C.cpf_candidato,
				C.nome_candidato,
				CD.código_da_candidatura,
				SUM(U.quantidade_de_votos_da_urna) AS QTD_VOTOS,
				SUM(B.valor_de_mercado_do_bem) / COUNT(C.cpf_candidato) AS SOMA
			FROM `Candidato` C
```

```
                        INNER JOIN `Candidatura` CD ON CD.cpf_candidato = C.cpf_candidato
                        INNER JOIN `Votação` V ON V.`código_da_candidatura` =
CD.código_da_candidatura
                        INNER JOIN `Votação_possui_urna` VPU ON VPU.código_da_votação =
V.código_da_votação
                        INNER JOIN `Urna` U ON U.código_da_urna = VPU.código_da_urna
                        INNER JOIN `Declaração_de_bens` DB ON DB.código_da_candidatura =
CD.código_da_candidatura
                        INNER JOIN `Bem` B ON B.código_da_declaração_de_bens =
DB.código_da_declaração
                        WHERE
                        V.situação_de_registro_de_candidatura_da_votação = 'apto' AND
                        VPU.ano = 2018 AND
                        U.município_da_urna = 'Campina Grande'
                GROUP BY C.cpf_candidato,
                                C.nome_candidato,
                                B.valor_de_mercado_do_bem,
                                CD.código_da_candidatura,
                                U.quantidade_de_votos_da_urna
        ) TEMP1
        GROUP BY TEMP1.cpf_candidato,
                TEMP1.nome_candidato,
                TEMP1.código_da_candidatura
        -- WHERE QTD_VOTOS > 500
) AS TEMP
        INNER JOIN `Doação_de_campanha` DC ON DC.código_da_candidatura =
TEMP.código_da_candidatura
        INNER JOIN `Mandato` M ON M.código_da_candidatura = TEMP.código_da_candidatura
        INNER JOIN `Gasto` G ON G.código_do_mandato = M.código_do_mandato
WHERE TEMP.QTD_VOTOS > 500
GROUP BY TEMP.cpf_candidato,
        TEMP.nome_candidato,
        TEMP.código_da_candidatura,
        TEMP.TOTAL;

-- C)
-- Liste o nome de todas as cidades e o respectivo total de votos no pleito de 2018 para o candidato
"João".

select u.município_da_urna as 'Cidades', SUM(u.quantidade_de_votos_da_urna) as 'Total de votos'
from urna as u, votação_possui_urna as vpu, votação as v, candidatura as ctra, candidato as c
where c.nome_candidato = 'João' and c.cpf_candidato = ctra.cpf_candidato and
ctra.código_da_candidatura = v.código_da_candidatura
and v.código_da_votação = vpu.código_da_votação and vpu.código_da_urna = u.código_da_urna
and vpu.ano = '2018'
group by u.município_da_urna;

-- D)
--  Selecione o total de doação partidária recebido pelos candidatos do partido ‘CDU’.

select f.nome_do_partido_ficha_de_filiação as 'Partido', SUM(d.valor_doado_doação) as 'Total
doado'
```

```
from Ficha_de_filiação as f, candidatura as ctra, Doação_de_campanha as d, Candidato as c
where f.nome_do_partido_ficha_de_filiação = 'CDU' and f.número_do_partido_ficha_de_filiação =
ctra.número_do_partido_ficha_de_filiação
and ctra.código_da_candidatura = d.código_da_candidatura and ctra.cpf_candidato =
c.cpf_candidato;

-- E)
-- Apresente uma relação dos 3 partidos que receberam maior número de verbas partidárias

SELECT f.nome_do_partido_ficha_de_filiação as 'Partido', f.número_do_partido_ficha_de_filiação as
'Número do partido' , SUM(d.`valor_doado_doação`) as "Verbas"
FROM `Doação_de_campanha` as d, `Ficha_de_filiação` as f, Candidatura as ctra
WHERE d.`código_da_candidatura` = ctra.`código_da_candidatura` and
ctra.número_do_partido_ficha_de_filiação = f.número_do_partido_ficha_de_filiação
GROUP BY f.nome_do_partido_ficha_de_filiação
ORDER BY SUM(d.`valor_doado_doação`) DESC
LIMIT 3;


-- F)
-- Relacione os 5 partidos mais bem votados na eleição de 2018 na Paraíba (considere apenas
candidatos cujo registros esteja apto).

SELECT f.nome_do_partido_ficha_de_filiação as 'Partido', f.número_do_partido_ficha_de_filiação as
'Número do partido'
FROM `Ficha_de_filiação` as f, Candidatura as ctra, `Votação` as v, `Votação_possui_urna` as vu,
Urna as u
WHERE v.`código_da_votação` = vu.`código_da_votação` and ctra.`código_da_candidatura` =
v.`código_da_candidatura`
and u.`código_da_urna` = vu.`código_da_urna` and vu.ano = 2018 and u.estado_da_urna = 'Paraíba'
and v.`situação_de_registro_de_candidatura_da_votação` = 'apto'
GROUP BY f.nome_do_partido_ficha_de_filiação
ORDER BY SUM(u.`quantidade_de_votos_da_urna`) DESC
LIMIT 5;

-- G)
-- Relacione os 10 candidatos mais ricos que foram eleitos na Paraíba em 2018 (considere apenas
candidatos cujo registros esteja apto)

SELECT
   TEMP.cpf_candidato AS CPF_CANDIDATO,
   TEMP.nome_candidato AS NOME_CANDIDATO,
   SUM(TEMP.SOMA) TOTAL
FROM (
   SELECT
      C.cpf_candidato,
      C.nome_candidato,
      -- B.valor_de_mercado_do_bem,
      SUM(B.valor_de_mercado_do_bem) / COUNT(C.cpf_candidato) AS SOMA
   FROM `Candidato` C
   INNER JOIN `Candidatura` CD ON CD.cpf_candidato = C.cpf_candidato
   INNER JOIN `Votação` V ON V.`código_da_candidatura` = CD.código_da_candidatura
```

```
   INNER JOIN `Votação_possui_urna` VPU ON VPU.código_da_votação = V.código_da_votação
   INNER JOIN `Urna` U ON U.código_da_urna = VPU.código_da_urna
   INNER JOIN `Declaração_de_bens` DB ON DB.código_da_candidatura =
CD.código_da_candidatura
   INNER JOIN `Bem` B ON B.código_da_declaração_de_bens = DB.código_da_declaração
   WHERE
   V.resultado_da_votação = 'eleito' AND
   V.situação_de_registro_de_candidatura_da_votação = 'apto' AND
   VPU.ano = 2018 AND
   U.estado_da_urna = 'Paraíba'
   GROUP BY C.cpf_candidato, C.nome_candidato, B.valor_de_mercado_do_bem
) AS TEMP
GROUP BY TEMP.cpf_candidato, TEMP.nome_candidato order by SUM(TEMP.SOMA) desc
LIMIT 10;

-- H)
-- Apresente uma lista de todas as cidades e seus respectivos quantitativos de votos nos candidatos
do partido ‘CDU’. Ordene pela quantidade de votos válidos.

select f.nome_do_partido_ficha_de_filiação as 'Nome', sum(u.quantidade_de_votos_da_urna) as
'Votos', u.município_da_urna as 'Cidade'
from urna as u, ficha_de_filiação as f, candidatura as ctra, votação as v, votação_possui_urna as vpu,
candidato as c
where f.nome_do_partido_ficha_de_filiação = 'CDU' and ctra.número_do_partido_ficha_de_filiação =
f.número_do_partido_ficha_de_filiação
and v.código_da_candidatura = ctra.código_da_candidatura and v.código_da_votação =
vpu.código_da_votação
and vpu.código_da_urna = u.código_da_urna AND ctra.cpf_candidato = c.cpf_candidato
group by u.município_da_urna order by sum(u.quantidade_de_votos_da_urna) desc;
```