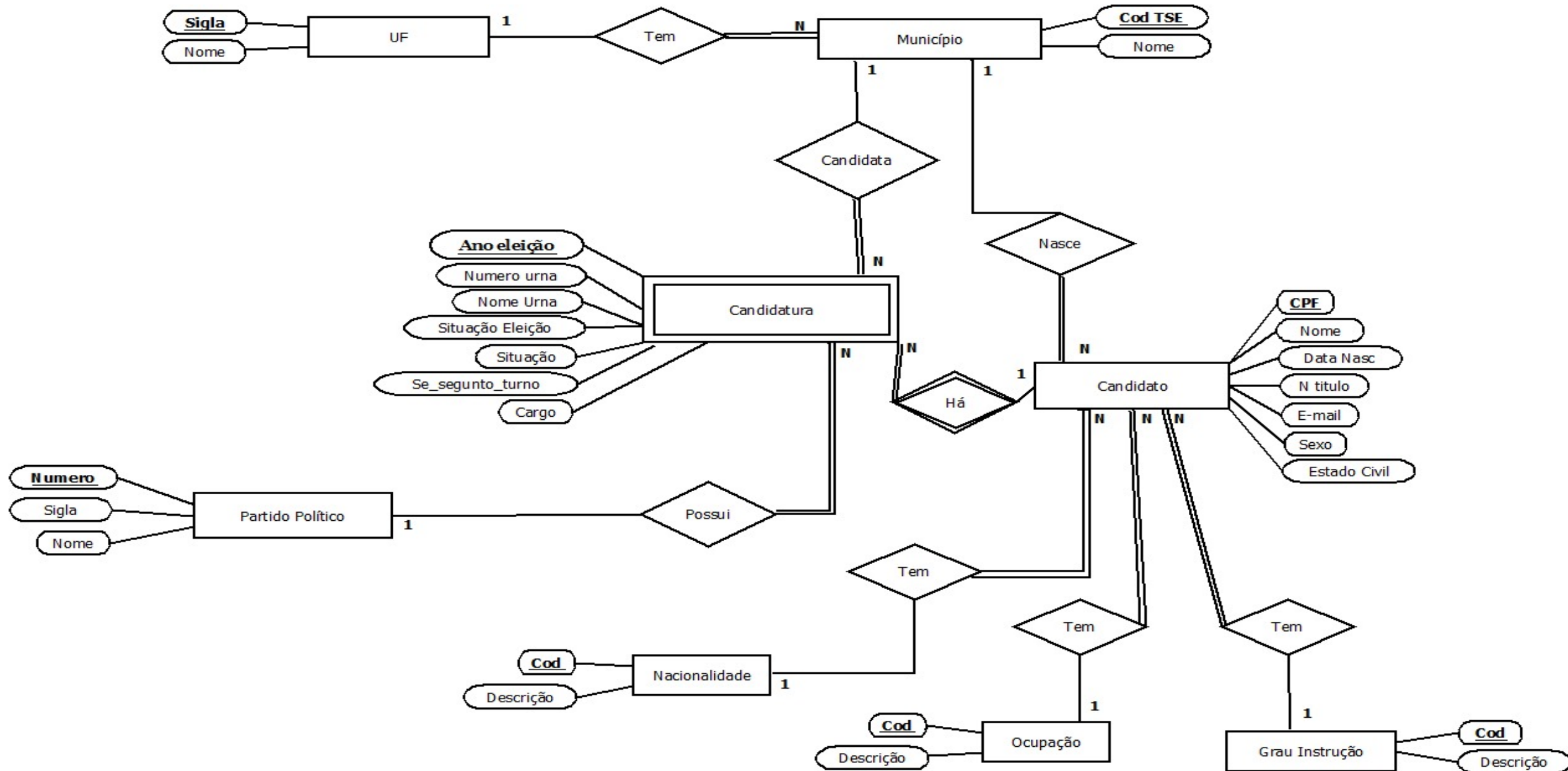


Marcito Campos
Rodrigo Hagstrom
Tássio Melo
Welder Luz

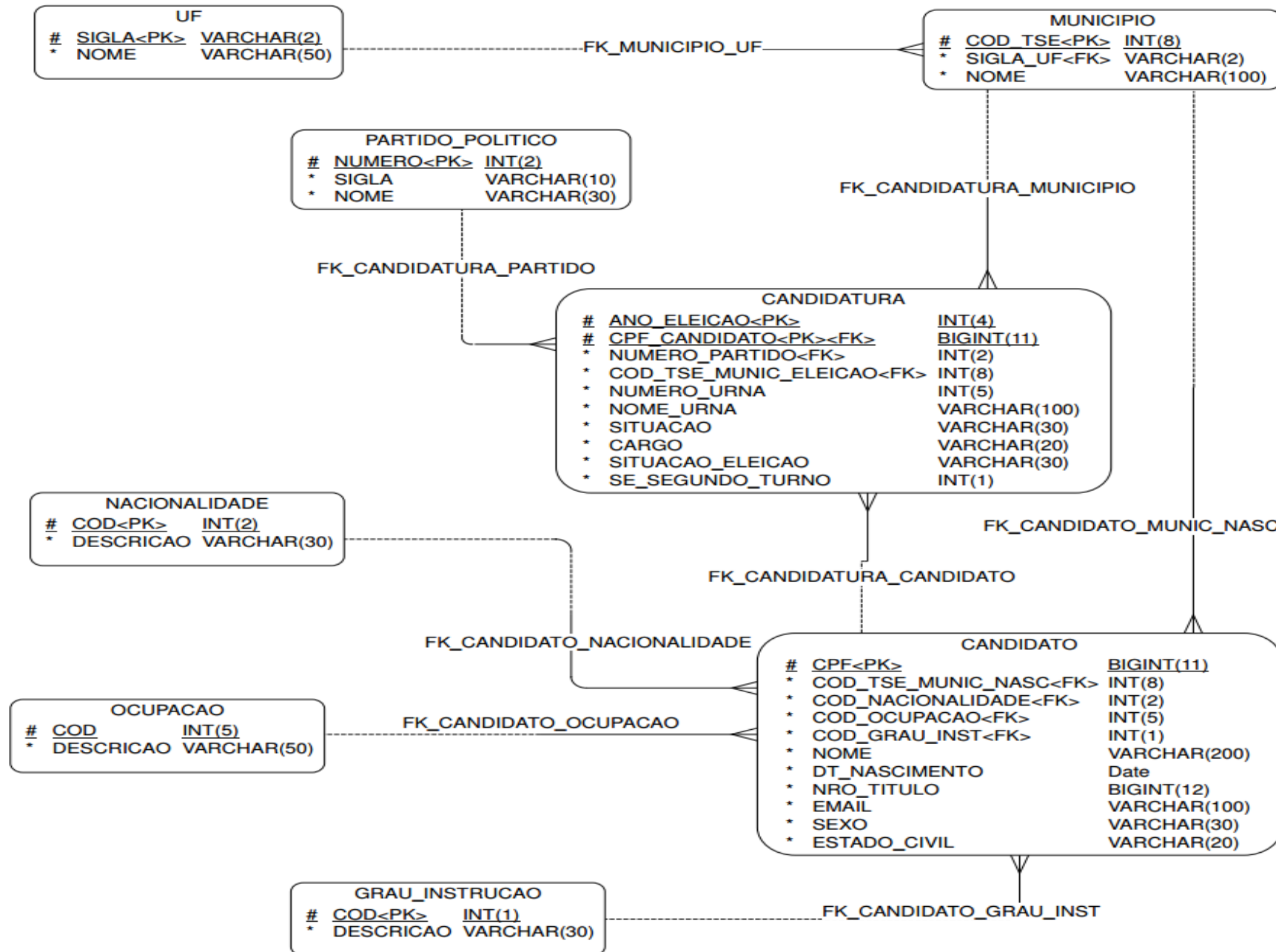
Tema

- Eleições Municipais – TSE
 - Perfil dos candidatos;
 - Ano: 2012;
 - Dados de 2016 ainda não estão em dados.gov.br

Modelo de Entidades e Relacionamentos



Modelo Relacional



Normalização 1/5

1º Forma Normal

CANDIDATURA

- * CPF<PK>
- * COD_TSE_MUNIC_NASC
- * NOME_MUNIC_NASC
- * SIGLA_UF_NASC
- * NOME
- * DT_NASCIMENTO
- * NRO_TITULO
- * EMAIL
- * SEXO
- * ESTADO_CIVIL
- * COD_TSE_MUNIC_ELEICAO
- * NOME_TSE_MUNIC_ELEICAO
- * SIGLA_UF_ELEICAO
- * ANO_ELEICAO<PK>
- * NUMERO_URNA
- * NOME_URNA
- * CARGO
- * SITUACAO
- * SE_ELEITO

Normalização 2/5

2ª Forma Normal

- **Chaves candidatas:** {CPF, ANO_ELEICAO}
- **Complementos da chave:** COD_TSE_MUNIC_NASC, NOME_MUNIC_NASC, SIGLA_UF_NASC, NOME, DT_NASCIMENTO, NRO_TITULO, EMAIL, SEXO, ESTADO_CIVIL, COD_TSE_MUNIC_ELEICAO, NOME_TSE_MUNIC_ELEICAO, SIGLA_UF_ELEICAO, NUMERO_URNA, NOME_URNA, CARGO, SITUACAO e SE_ELEITO.
- **Análise de dependências funcionais:**
 - $(CPF, ANO_ELEICAO) \rightarrow COD_TSE_MUNIC_ELEICAO, NOME_TSE_MUNIC_ELEICAO, SIGLA_UF_ELEICAO, NUMERO_URNA, NOME_URNA, CARGO, SITUACAO, SE_ELEITO;$
 - $(CPF, ANO_ELEICAO) \nrightarrow COD_TSE_MUNIC_NASC, NOME_MUNIC_NASC, SIGLA_UF_NASC, NOME, DT_NASCIMENTO, NRO_TITULO, EMAIL, SEXO, ESTADO_CIVIL;$
 - $(CPF) \rightarrow COD_TSE_MUNIC_NASC, NOME_MUNIC_NASC, SIGLA_UF_NASC, NOME, DT_NASCIMENTO, NRO_TITULO, EMAIL, SEXO, ESTADO_CIVIL.$

Normalização 3/5

▪ Modelo Relacional em 2FN:

CANDIDATURA

CPF<PK><FK>
* COD_TSE_MUNIC_ELEICAO
* NOME_TSE_MUNIC_ELEICAO
* SIGLA_UF_ELEICAO
ANO_ELEICAO<PK>
* NUMERO_URNA
* NOME_URNA
* CARGO
* SITUACAO
* SE_ELEITO

CANDIDATO

CPF<PK>
* COD_TSE_MUNIC_NASC
* NOME_MUNIC_NASC
* SIGLA_UF_NASC
* NOME
* DT_NASCIMENTO
* NRO_TITULO
* EMAIL
* SEXO
* ESTADO_CIVIL

Normalização 4/5

3ª Forma Normal

- **Análise das dependências transitivas:**

- Tabela CANDIDATURA:

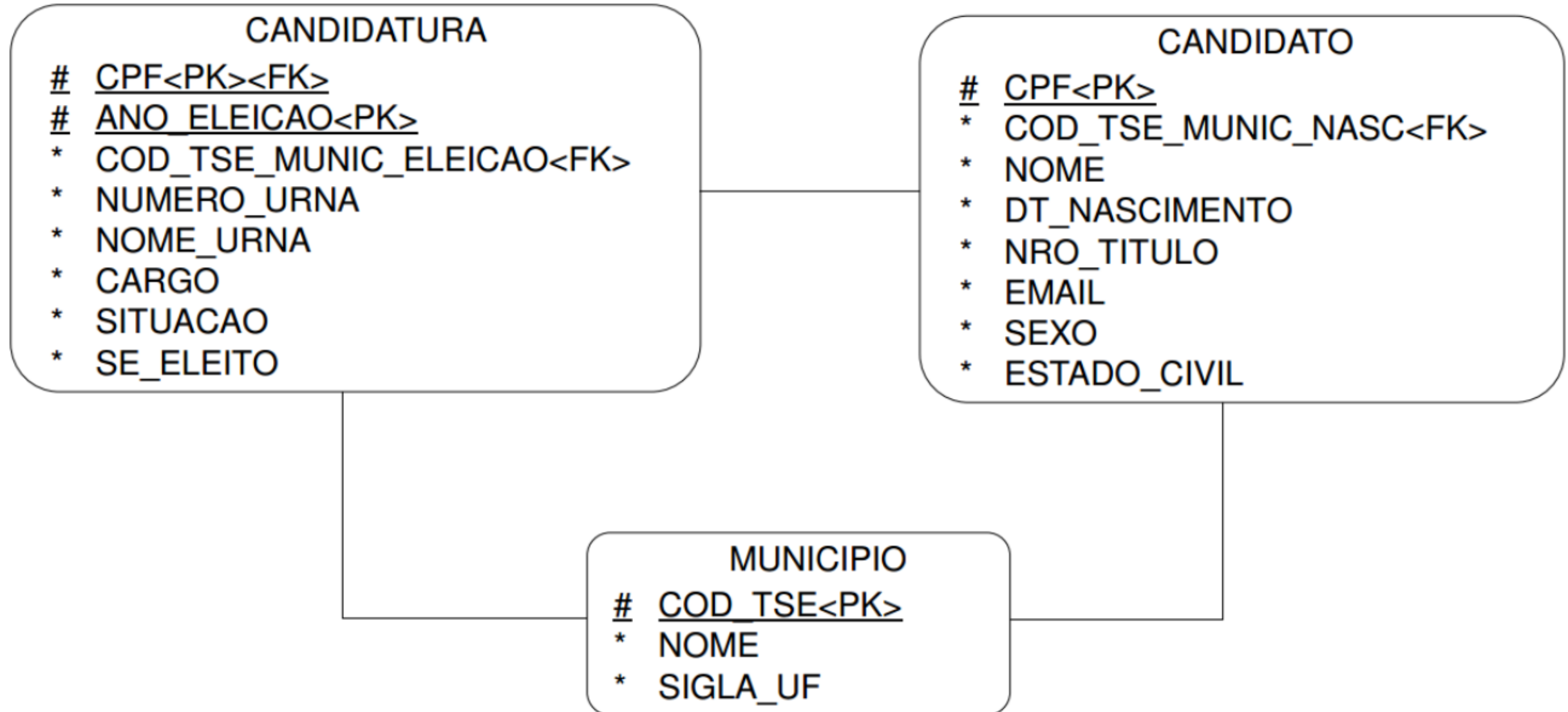
- $\text{COD_TSE_MUNIC_ELEICAO} \rightarrow \text{NOME_TSE_MUNIC_ELEICAO}, \text{SIGLA_UF_ELEICAO}.$
 - Nenhum outro atributo não chave determina funcionalmente outro também não chave.

- Tabela CANDIDATO:

- $\text{COD_TSE_MUNIC_NASC} \rightarrow \text{NOME_MUNIC_NASC}, \text{SIGLA_UF_NASC}.$
 - Nenhum outro atributo não chave determina funcionalmente outro também não chave.

Normalização 5/5

- Modelo Relacional em 3FN:



SGBD

- Maria DB;
 - Fork do MySQL após aquisição pela Oracle;
- Motivos da escolha:
 - Simplicidade;
 - Licença GLP.

Criação do banco de dados (1/2)

```
CREATE DATABASE projeto_final;
CREATE TABLE UF (
    SIGLA VARCHAR(2) PRIMARY KEY NOT NULL,
    NOME VARCHAR(50) NOT NULL
);
CREATE TABLE MUNICIPIO (
    COD_TSE INT(8) PRIMARY KEY NOT NULL,
    SIGLA_UF VARCHAR(2) NOT NULL,
    NOME VARCHAR(100) NOT NULL,
    CONSTRAINT FK_MUNICIPIO_UF
        FOREIGN KEY (SIGLA_UF) REFERENCES UF(SIGLA)
);
CREATE TABLE PARTIDO_POLITICO (
    NUMERO INT(2) PRIMARY KEY NOT NULL,
    SIGLA VARCHAR(10) NOT NULL,
    NOME VARCHAR(50) NOT NULL
);
CREATE TABLE GRAU_INSTRUCAO (
    COD INT(1) PRIMARY KEY NOT NULL,
    DESCRICAO VARCHAR(30) NOT NULL
);
CREATE TABLE NACIONALIDADE (
    COD INT(2) PRIMARY KEY NOT NULL,
    DESCRICAO VARCHAR(50) NOT NULL
);
CREATE TABLE OCUPACAO (
    COD INT(5) PRIMARY KEY NOT NULL,
    DESCRICAO VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE CANDIDATO (
    CPF BIGINT(11) PRIMARY KEY NOT NULL,
    COD_TSE_MUNIC_NASC INT(8) NOT NULL,
    COD_NACIONALIDADE INT(2) NOT NULL,
    COD_OCUPACAO INT(5) NOT NULL,
    COD_GRAU_INST INT(1) NOT NULL,
    NOME VARCHAR(200) NOT NULL,
    DT_NASCIMENTO DATE NOT NULL,
    NRO_TITULO BIGINT(12) NOT NULL,
    EMAIL VARCHAR(100),
    SEXO VARCHAR(30) NOT NULL,
    ESTADO_CIVIL VARCHAR(50) NOT NULL,
    CONSTRAINT FK_CANDIDATO_MUNIC_NASC
        FOREIGN KEY (COD_TSE_MUNIC_NASC) REFERENCES MUNICIPIO(COD_TSE),
    CONSTRAINT FK_CANDIDATO_NACIONALIDADE
        FOREIGN KEY (COD_NACIONALIDADE) REFERENCES NACIONALIDADE(COD),
    CONSTRAINT FK_CANDIDATO_OCUPACAO
        FOREIGN KEY (COD_OCUPACAO) REFERENCES OCUPACAO(COD),
    CONSTRAINT FK_CANDIDATO_GRAU_INST
        FOREIGN KEY (COD_GRAU_INST) REFERENCES GRAU_INSTRUCAO(COD)
);
```

Criação do banco de dados (2/2)

```
CREATE TABLE CANDIDATURA (  
    CPF_CANDIDATO BIGINT(11) NOT NULL,  
    ANO_ELEICAO INT(4) NOT NULL,  
    NUMERO_PARTIDO INT(2) NOT NULL,  
    COD_TSE_MUNIC_ELEICAO INT(8) NOT NULL,  
    NUMERO_URNA INT(5) NOT NULL,  
    NOME_URNA VARCHAR(100) NOT NULL,  
    SITUACAO VARCHAR(50) NOT NULL,  
    CARGO VARCHAR(20) NOT NULL,  
    SITUACAO_ELEICAO VARCHAR(30) NOT NULL,  
    SE_SEGUNDO_TURNO INT(1) NOT NULL,  
    CONSTRAINT PK_CANDIDATURA  
        PRIMARY KEY (CPF_CANDIDATO, ANO_ELEICAO),  
    CONSTRAINT FK_CANDIDATURA_CANDIDATO  
        FOREIGN KEY (CPF_CANDIDATO) REFERENCES CANDIDATO(CPF),  
    CONSTRAINT FK_CANDIDATURA_PARTIDO  
        FOREIGN KEY (NUMERO_PARTIDO) REFERENCES PARTIDO_POLITICO(NUMERO),  
    CONSTRAINT FK_CANDIDATURA_MUNICIPIO  
        FOREIGN KEY (COD_TSE_MUNIC_ELEICAO) REFERENCES MUNICIPIO(COD_TSE),  
    UNIQUE(CPF_CANDIDATO, ANO_ELEICAO, COD_TSE_MUNIC_ELEICAO)  
);
```

Procedure (1/3)

```
DELIMITER //
CREATE PROCEDURE incluir_ufs(OUT retorno char(100))
BEGIN
    DECLARE NRO_UFS_CAD INT;
    SELECT COUNT(SIGLA) INTO NRO_UFS_CAD FROM UF;
    CASE
        WHEN NRO_UFS_CAD > 0 THEN
            SET retorno = 'Já havia UF cadastrada, inclusão NÃO realizada';
        ELSE
            insert into UF(SIGLA, NOME) VALUES ('AC', 'Acre');
            insert into UF(SIGLA, NOME) VALUES ('AL', 'Alagoas');
            insert into UF(SIGLA, NOME) VALUES ('AM', 'Amazonas');
            (...)
            SET retorno = 'Não havia UF cadastrada, inclusão realizada com sucesso';
        END CASE;
    END//
```

Procedure (2/3)

```
MariaDB [projeto_final]> select count(*) from UF;
```

```
+-----+  
| count(*) |  
+-----+  
|          0 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> call incluir_ufs(@retorno);  
Query OK, 1 row affected (0.15 sec)
```

```
MariaDB [projeto_final]> select @retorno;
```

```
+-----+  
| @retorno |  
+-----+  
| Não havia UF cadastrada, inclusão realizada com sucesso |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> select count(*) from UF;
```

```
+-----+  
| count(*) |  
+-----+  
|          28 |  
+-----+
```

```
1 row in set (0.00 sec)
```

Procedure (3/3)

```
MariaDB [projeto_final]> select count(*) from UF;
```

```
+-----+  
| count(*) |  
+-----+  
|      28 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> call incluir_ufs(@retorno);  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [projeto_final]> select @retorno;
```

```
+-----+  
| @retorno |  
+-----+  
| Já havia UF cadastrada, inclusão NÃO realizada |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> select count(*) from UF;
```

```
+-----+  
| count(*) |  
+-----+  
|      28 |  
+-----+
```

```
1 row in set (0.00 sec)
```

Trigger (1/3)

```
DELIMITER //
CREATE TRIGGER validar_idade
BEFORE INSERT on CANDIDATURA FOR EACH ROW
BEGIN
    DECLARE DT_NASC DATE;
    DECLARE MSG VARCHAR(255);

    SELECT c.DT_NASCIMENTO
    INTO DT_NASC
    FROM CANDIDATO c
    WHERE c.CPF = New.CPF_CANDIDATO;

    CASE
        WHEN New.CARGO = 'PREFEITO'
        AND DATEDIFF(concat(New.ANO_ELEICAO, '-10-01'), DT_NASC) < 7665
        THEN
            SET MSG = "ERRO: Candidato a prefeito precisa possuir 21 anos.";
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
        ELSE BEGIN END;
    END CASE;
END; //
```


Trigger (2/3)

```
MariaDB [projeto_final]> SELECT DT_NASCIMENTO FROM CANDIDATO WHERE CPF=13911893450;
```

```
+-----+  
| DT_NASCIMENTO |  
+-----+  
| 1991-03-25    |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> INSERT INTO CANDIDATURA (CPF_CANDIDATO, ANO_ELEICAO, NUMERO_PARTIDO, COD_TSE_MUNIC_ELEICAO  
,  
    NUMERO_URNA, NOME_URNA, SITUACAO, CARGO, SITUACAO_ELEICAO, SE_SEGUNDO_TURNO)  
    -> VALUES (13911893450, 2008, 22, 10375, 22222, 'Xavier Neto', 'DEFERIDO', 'PREFEITO', 'ELEITO', 0);
```

```
ERROR 1644 (45000): ERRO: Candidato a prefeito precisa possuir 21 anos.
```

```
MariaDB [projeto_final]> SELECT COUNT(*) FROM CANDIDATURA WHERE CPF_CANDIDATO=13911893450 AND ANO_ELEICAO=2008;
```

```
+-----+  
| COUNT(*) |  
+-----+  
|         0 |  
+-----+
```

```
1 row in set (0.00 sec)
```

Trigger (3/3)

```
MariaDB [projeto_final]> SELECT DT_NASCIMENTO FROM CANDIDATO WHERE CPF=13911893450;
```

```
+-----+  
| DT_NASCIMENTO |  
+-----+  
| 1991-03-25    |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
MariaDB [projeto_final]> INSERT INTO CANDIDATURA (CPF_CANDIDATO, ANO_ELEICAO, NUMERO_PARTIDO, COD_TSE_  
MUNIC_ELEICAO, NUMERO_URNA, NOME_URNA, SITUACAO, CARGO, SITUACAO_ELEICAO, SE_SEGUNDO_TURNO)  
-> VALUES (13911893450, 2012, 22, 10375, 22222, 'Xavier Neto', 'DEFERIDO', 'PREFEITO', 'ELEITO', 0  
);
```

```
Query OK, 1 row affected (0.01 sec)
```

```
MariaDB [projeto_final]> SELECT COUNT(*) FROM CANDIDATURA WHERE CPF_CANDIDATO=13911893450 AND ANO_ELEICAO=2012;
```

```
+-----+  
| COUNT(*) |  
+-----+  
|         1 |  
+-----+
```

```
1 row in set (0.00 sec)
```

View / Consulta (1/4)

View / Consulta (2/4)

View / Consulta (3/4)

View / Consulta (4/4)

ETL

- Aplicação Java
 - Seleção de arquivos;
 - Individualmente;
 - Diretório completo.
 - Conversão de linha de arquivo em objeto;
 - Persistência no banco de dados por meio do Hibernate.

OBRIGADO!!!