Dados Abertos
RS - Votação por Seção Eleitoral 2020

Discentes: Iago Mendes Nogueira e Adriel Luan da Silveira
Docente: Maicon Bernardino da Silveira

[1]Engenharia de Software – Universidade Federal do Pampa - Alegrete (Unipampa)
97546-550 – Alegrete – RS – Brazil

Abstract. *This file describes the database process that was proposed by the teacher, so that it is consistent with the issues discussed below. It contains an introduction to the topic, the generated files (Standardization, Logical/Relational Project, Conceptual Project and Physical Project), Lessons learned and the References that were used.*

Resumo. *Este arquivo descreve o processo de banco de dados que foi proposto pelo docente, de maneira que esteja coerente com os assuntos abordados a seguir. O mesmo contém uma introdução ao tema, os arquivos gerados(Normalização, Projeto Lógico/Relacional, Projeto Conceitual e Projeto Físico), Lições aprendidas e as Referências que foram utilizadas.*

1. Introdução

1.1. Contextualização do Problema

Visando uma melhoria no conjunto de dados do governo federal, foi elaborado um banco de dados que visa auxiliar na consulta sobre os resultados das eleições do estado do Rio Grande do Sul no ano de 2020.

1.2. Descrição

O modelo de Banco de Dados elaborado pelos discentes, possui os processos propostos pelo docente, buscando um projeto coerente de fácil entendimento e que atinja todas as expectativas esperadas.

1.3. Justificativa

A escolha do respectivo tema, se deu pelo fato de que a disponibilização dos resultados por parte do governo federal é de difícil navegação e compreensão. Impossibilitando que alguém com interesse no tema, não consiga se adequar ao arquivo.

2. Normalização

O processo de Normalização em banco de dados consiste em seguir um conjunto de regras que visa a organização do projeto, eliminando redundância de dados aumentando a integridade e o desempenho dos dados. A normalização dos dados parte da análise das colunas e os relacionamento entre as entidades.[Desconhecido 2021b],[Rodrigues 2014]

2.1. ÑN

A primeira parte da Normalização é constituída pela **Não-Normalizada(ÑN)**, onde a tabela ainda se encontra em sua originalidade sem alterações, onde é usado no esquema um outro tipo de notação.

Tabela: Eleições 2020

- **CD-Tipo-Eleição**
- **CD-Eleição**
- **NM-Tipo-Eleição**
- **NR-Turno**
- **DS-Eleição**
- **TP-Abrangência**
- **SG-UF**
- **SG-UE**
- **NM-UE**
- **NR-Zona**
- **NR-Seção**
- **NR-Votável**
- **CD-Cargo**
- **DS-Cargo**
- **NM-Votável**
- **QT-Votos**
- **NR-Local-Votação**
- **ANO-Eleição**
- **DT-Geração**
- **HH-Geração**
- **DT-Eleição**

Figure 1. Esquema ÑN

2.2. Forma Normal

A forma normal é um conjunto de regras com critérios práticos de simplificação de tabelas contendo 4 tipos de formas normais.[Kolb 2017]

2.2.1. Primeira Forma Normal(1FN)

A tabela se encontra na primeira forma normal quando não possui nenhum domínio multivalorado. Aplicando as regras para a passagem 1FN gerou-se a seguinte normalização.

Figure 2. Esquema 1FN

2.2.2. Segunda Forma Normal(2FN)

Uma tabela encontra-se na segunda forma normal, quando, além de estar na 1FN, não contém dependências parciais. Aplicando as regras para a passagem 2FN gerou-se a seguinte normalização.

Figure 3. Esquema 2FN

2.2.3. Terceira Forma Normal(3FN)

Uma tabela encontra-se na terceira forma normal, quando, além de estar na 2FN, não contém dependências transitivas. Aplicando as regras para a passagem 3FN gerou-se a seguinte normalização.

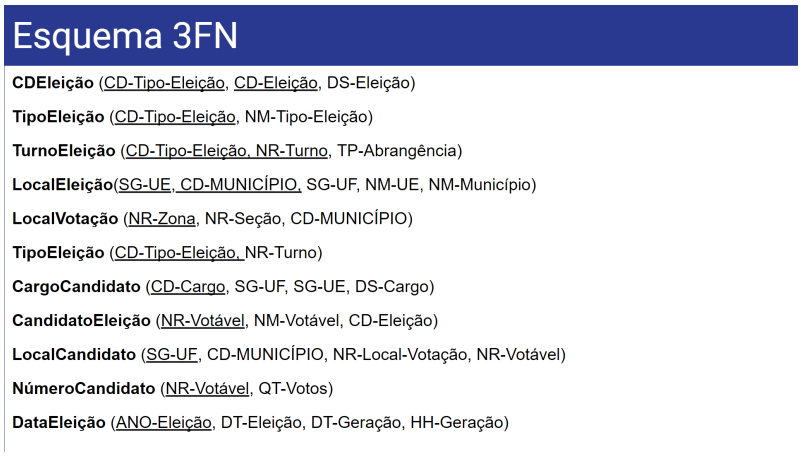

Figure 4. Esquema 3FN

2.2.4. Quarta Forma Normal(4FN)

Uma tabela encontra-se na quarta forma normal, quando, além de estar na 3FN, não contém mais de uma dependência multivalorada. Aplicando as regras para a passagem 4FN gerou-se a seguinte normalização.

Figure 5. Esquema 4FN

3. Projeto Lógico/Relacional

Seguindo os mesmos passos que foram executados para a elaboração do modelo lógico, foi realizado o modelo relacional, utilizando a plataforma do ”Eclipse” para o processo ERText.[Desconhecido 2021a]

```
Generate All;

Domain TemplateModel;

Entities {

    CdEleicao {
        cdEleicao int isIdentifier,
        cdTipoEleicao int
    }

    NomeEleicao {
        dsEleicao string isIdentifier,
        nmTipoEleicao string
    }

    TurnoEleicao {
        nrTurno int isIdentifier,
        tpAbrengecia string
    }

    LocalEleicao {
        sgUf string isIdentifier,
        sgUe int
    }

    NomeLocalEleicao {
        nmUe string isIdentifier,
        nmMunicipio string
    }

    LocalVotacao {
        cdMunicipio int isIdentifier,
        nmZona int,
        nrSecao int
    }
```

Figure 6. Modelo Lógico ERText parte 1

```
    TipoEleicao {
        cdTipoEleicao int isIdentifier,
        cdEleicao int
    }

    CargoCandidato {
        cdCargo int isIdentifier,
        dsCargo string
    }

    NumeroCandidato {
        nrVotavel int isIdentifier,
        nmVotavel string
    }

    VotosCandidato {
        qtVotos int isIdentifier
    }

    DataEleicao {
        anoEleicao dateTime isIdentifier,
        dtEleicao dateTime
    }

    DataGeracao {
        dtGEracao dateTime isIdentifier,
        hhGeracao dateTime
    }
};
```

Figure 7. Modelo Lógico ERText parte 2

```
Relationships {
        TipoEleicao [TipoEleicao (1:N) relates (1:N) CargoCandidato] {cdCargo int}
        CargoCandidato [CargoCandidato (1:1) relates (1:1) LocalEleicao]{sgUf string}
        NumeroCandidato [NumeroCandidato (1:1) relates (1:N) CargoCandidato]{cdCargo int}
        LocalCandidato [LocalEleicao (1:N) relates (1:N) LocalVotacao] {cdMunicipio int,sgUf string}
        LocalCandidatol [LocalCandidato (1:N) relates (1:N) CargoCandidato] {cdCargo int}
        VotosCandidato [TurnoEleicao (1:1) relates (1:N) CdEleicao] {nrTurno int,cdEleicao int}
        DataEleicao [DataEleicao (1:1) relates (1:1) CdEleicao] {cdEleicao int}
        DataGeracao [DataGeracao (1:1) relates (1:N) CdEleicao] {cdEleicao int}
};
```

Figure 8. Modelo Lógico ERText parte 3

4. Projeto Conceitual

Após a elaboração da Normalização, foi desenvolvido o modelo lógico, contendo as entidades e relacionamentos em forma de diagrama [Desconhecido 2021a]

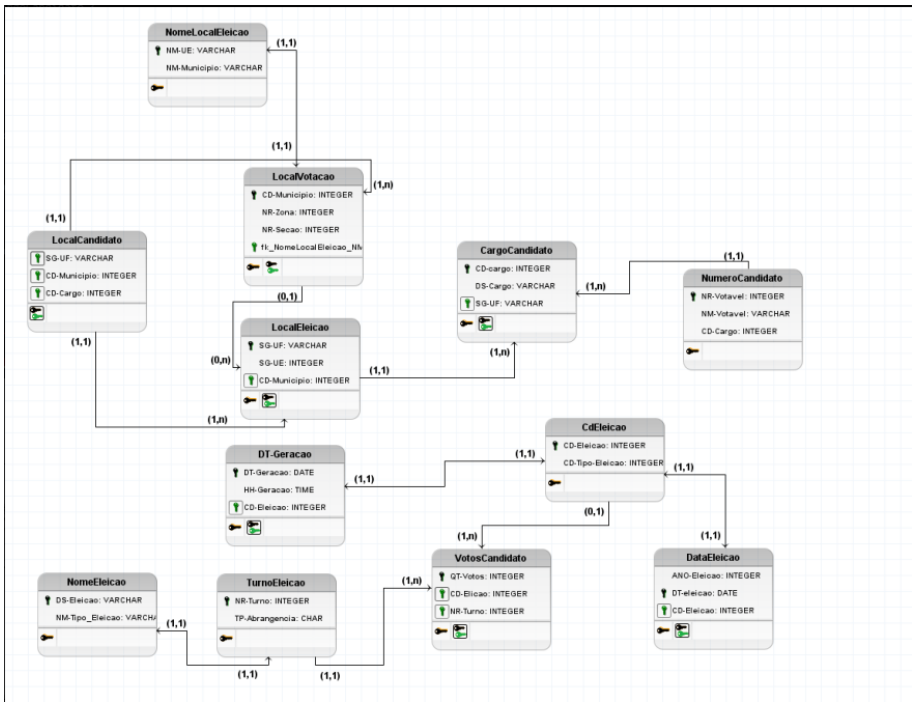

Figure 9. Modelo Conceitual

5. Projeto Físico

O Projeto Físico colocamos em prática a modelagem em SQL. Pela plataforma do PostgreSQL [PostgreeSQL 2022] foi elaborado a criação das tabelas (DDL) e as consultas (DML), utilizando os dados abertos disponibilizados no site do Tribunal Superior Eleitoral[Justiça 2021], onde é abordado o tema ”Resultados por seção eleitoral” do ano de 2020 no estado do Rio Grande do Sul.[Sanches 2005]

5.1. DDL

```
CREATE TABLE IF NOT EXISTS eleicoes.cargocandidato
(
    "cd-cargo" integer[] NOT NULL,
```

```
    "ds-cargo" character varying[] COLLATE pg_catalog."default" NOT NUL
    "sg-uf" character varying[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT cargocandidato_pkey PRIMARY KEY ("cd-cargo"),
    CONSTRAINT fk_loccarg FOREIGN KEY ("sg-uf")
        REFERENCES eleicoes.localeleicao ("sg-uf") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.cargocandidato
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.cargocandidato TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.cd_eleicao
(
    "cd-eleicao" integer[] NOT NULL,
    "cd-tipo-eleicao" integer[] NOT NULL,
    CONSTRAINT cdeleicao_pkey PRIMARY KEY ("cd-eleicao")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.cd_eleicao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.cd_eleicao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.dataeleicao
(
    "ano-eleicao" integer[] NOT NULL,
    "dt-eleicao" integer[] NOT NULL,
    "cd-eleicao" integer[] NOT NULL,
    CONSTRAINT dataeleicao_pkey PRIMARY KEY ("ano-eleicao"),
    CONSTRAINT fk_cddata FOREIGN KEY ("cd-eleicao")
        REFERENCES eleicoes.cd_eleicao ("cd-eleicao") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;
CREATE TABLE IF NOT EXISTS eleicoes.datageracao
(
    "dt-geracao" integer[] NOT NULL,
    "hh-geracao" integer[] NOT NULL,
    "cd-eleicao" integer[] NOT NULL,
```

```
    CONSTRAINT datageracao_pkey PRIMARY KEY ("dt-geracao"),
    CONSTRAINT fk_cddatag FOREIGN KEY ("cd-eleicao")
        REFERENCES eleicoes.cd_eleicao ("cd-eleicao") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.datageracao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.datageracao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.localeleicao
(
    "sg-uf" character varying[] COLLATE pg_catalog."default" NOT NULL,
    "sg-ue" integer[] NOT NULL,
    "cd-municipio" integer[] NOT NULL,
    CONSTRAINT localeleicao_pkey PRIMARY KEY ("sg-uf"),
    CONSTRAINT fk_loctipo FOREIGN KEY ("cd-municipio")
        REFERENCES eleicoes.localvotacao ("cd-municipio") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.localeleicao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.localeleicao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.localvotacao
(
    "nr-zona" integer[] NOT NULL,
    "nr-secao" integer[] NOT NULL,
    "cd-municipio" integer[] NOT NULL,
    CONSTRAINT localvotacao_pkey PRIMARY KEY ("cd-municipio")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.localvotacao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.localvotacao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.nomeeleicao
```

```
(
    "ds-eleicao" character varying[] COLLATE pg_catalog."default" NOT N
    "nm-tipo-eleicao" character varying[] COLLATE pg_catalog."default" 
    CONSTRAINT nomeeleicao_pkey PRIMARY KEY ("ds-eleicao")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.nomeeleicao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.nomeeleicao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.nomelocalvotacao
(
    "nm-ue" character varying[] COLLATE pg_catalog."default" NOT NULL,
    "nm-municipio" character varying[] COLLATE pg_catalog."default" NOT
    CONSTRAINT nomelocalvotacao_pkey PRIMARY KEY ("nm-ue")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.nomelocalvotacao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.nomelocalvotacao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.numerocandidato
(
    "nr-votavel" integer[] NOT NULL,
    "nm-votavel" character varying[] COLLATE pg_catalog."default" NOT N
    "cd-cargo" integer[] NOT NULL,
    CONSTRAINT numerocandidato_pkey PRIMARY KEY ("nr-votavel"),
    CONSTRAINT fk_numcarg FOREIGN KEY ("cd-cargo")
        REFERENCES eleicoes.cargocandidato ("cd-cargo") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.numerocandidato
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.numerocandidato TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.tipoeleicao
(
    "cd-tipo-eleicao" integer[] NOT NULL,
```

```
    "cd-eleicao" integer[] NOT NULL,
    "cd-cargo" integer[] NOT NULL,
    CONSTRAINT tipoeleicao_pkey PRIMARY KEY ("cd-tipo-eleicao"),
    CONSTRAINT fk_tipcarg FOREIGN KEY ("cd-cargo")
        REFERENCES eleicoes.cargocandidato ("cd-cargo") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.tipoeleicao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.tipoeleicao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.turnoeleicao
(
    "nr-turno" integer[] NOT NULL,
    "tp-abrangecia" character varying[] COLLATE pg_catalog."default" NO
    CONSTRAINT turnoeleicao_pkey PRIMARY KEY ("nr-turno")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS eleicoes.turnoeleicao
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.turnoeleicao TO postgres;
CREATE TABLE IF NOT EXISTS eleicoes.votoscandidato
(
    "qt-votos" integer[] NOT NULL,
    "nr-turno" integer[] NOT NULL,
    "cd-eleicao" integer[] NOT NULL,
    CONSTRAINT votoscandidato_pkey PRIMARY KEY ("qt-votos"),
    CONSTRAINT fk_cddatag FOREIGN KEY ("cd-eleicao")
        REFERENCES eleicoes.cd_eleicao ("cd-eleicao") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT fk_turvot FOREIGN KEY ("nr-turno")
        REFERENCES eleicoes.turnoeleicao ("nr-turno") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;
```

```
ALTER TABLE IF EXISTS eleicoes.votoscandidato
    OWNER to postgres;

GRANT ALL ON TABLE eleicoes.votoscandidato TO postgres;
```

5.2. DML

```
SELECT "nr-votavel" FROM eleicoes.numerocandidato
UNION
SELECT "sg-ue" FROM eleicoes.localeleicao
SELECT "nm-votavel" FROM eleicoes.numerocandidato
UNION
SELECT "sg-uf" FROM eleicoes.localeleicao
SELECT "cd-cargo" FROM eleicoes.cargocandidato
UNION
SELECT "cd-municipio" FROM eleicoes.localvotacao
SELECT "cd-eleicao" FROM eleicoes.cdeleicao
UNION
SELECT "ano-eleicao" FROM eleicoes.dataeleicao
SELECT "nm-votavel" FROM eleicoes.numerocandidato
SELECT  "cd-eleicao" FROM eleicoes.cd_eleicao WHERE "cd-tipo-eleicao" =
SELECT "nr-votavel" FROM eleicoes.numerocandidato
UNION
SELECT "sg-ue" FROM eleicoes.localeleicao
SELECT * FROM eleicoes.localvotacao
ORDER BY "cd-municipio" ASC
SELECT * FROM eleicoes.numerocandidato
ORDER BY "nr-votavel" ASC
SELECT * FROM eleicoes.tipoeleicao
ORDER BY "cd-tipo-eleicao" ASC
```

6. Lições Aprendidas

Podemos concluir que o aprendizado para ambos os discentes, foi o contato com um projeto de banco de dados, desde o processo inicial, até modelar os dados abertos em um modelo físico em SQL(DDL e DML). Por ser um conteúdo que nunca tivemos contato, passamos por algumas dificuldades no desenvolver do projeto, que foram superadas por algumas video aulas, slides e artigos.[Maicon Bernardino 2022]

References

Desconhecido (2021a). ”modelagem de dados: Modelo conceitual, modelo lÓgico e fÍsico”.

Desconhecido (2021b). ”normalização”.

Justiça, E. (16 de agosto de 2021). ”rs - votação por seção eleitoral - 2020”.

Kolb, J. J. (julho 21, 2017). ”formas normais”.

Maicon Bernardino, S. (2022). ”slides de banco de dados”.

PostgreeSQL, d. (1996-2022). ”postgresql documentation”.

Rodrigues, L. (2014). ”normalização de dados”.

Sanches, A. R. (14/04/2005). ”fundamentos de armazenamento e manipulação de dados”.