

# SQL para criação de um Data Warehouse dentro do PostgreSQL

Para criar um Data Warehouse dentro do PostgreSQL foi utilizado esse script com suas devidas tabelas dimensões, tabela fato, criação de chaves estrangeiras fora da tabela fato por boa prática de código SQL e criação de indexes para otimização de consultas e processo de carga da tabela fato.

## Estrutura do Banco de Dados

### Tabelas de Dimensões

1. **Dimensão Candidato:**
  - Esta dimensão armazena informações sobre os candidatos, incluindo dados como número de inscrição, faixa etária, sexo, etnia, estado civil, nacionalidade, status de conclusão e se é treineiro
2. **Dimensão Localidade:**
  - Nesta dimensão, são registradas informações sobre a localidade dos candidatos, como UF (Unidade Federativa) e o nome do município.
3. **Dimensão Prova:**
  - Esta dimensão contém dados relacionados à presença dos candidatos nas provas, incluindo informações sobre a presença em cada uma das provas (Ciências da Natureza, Ciências Humanas, Linguagens e Códigos, e Matemática).
4. **Dimensão Escola:**
  - Na dimensão Escola, são registradas informações sobre as escolas dos candidatos, como tipo de escola (pública, privada, etc.) e localização da escola.

### Tabela de Fato Notas

- A tabela de Fato Notas armazena os dados relacionados às notas dos candidatos, incluindo a nota em cada uma das provas, nota da redação e notas de competências específicas.

### Chaves Estrangeiras

- Foram estabelecidas chaves estrangeiras na tabela de Fato Notas para garantir a integridade referencial com as dimensões relacionadas (Candidato, Escola, Prova e Localidade).

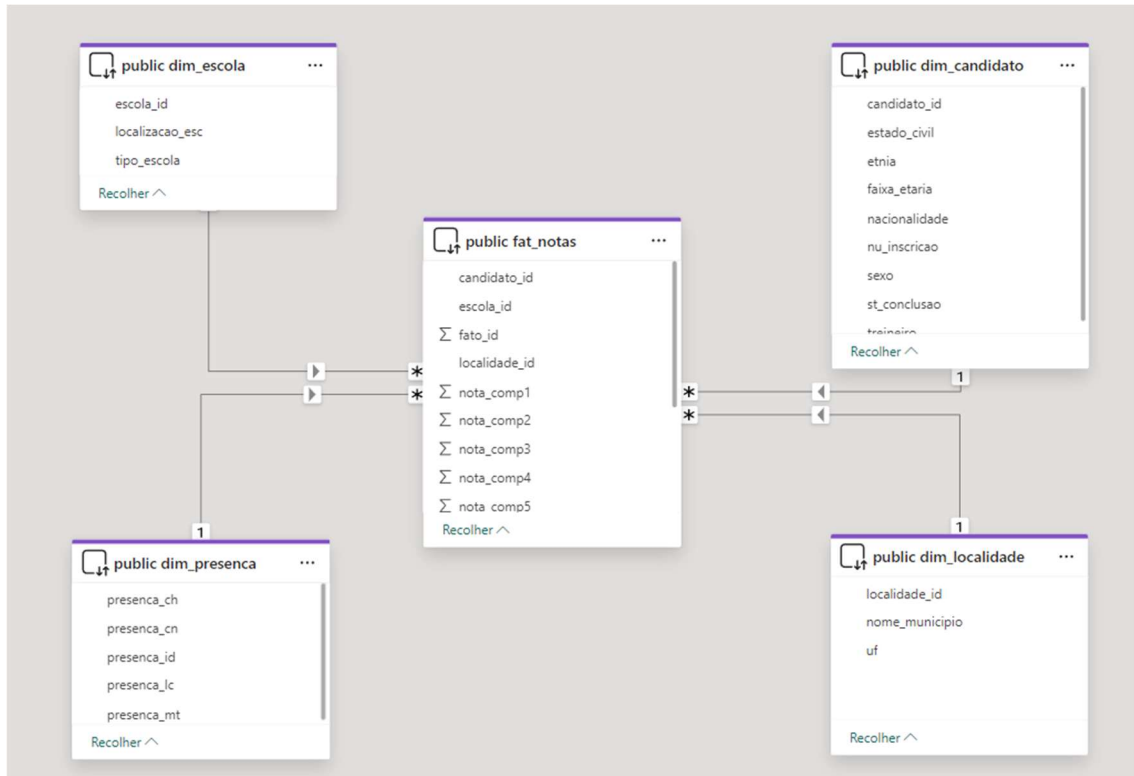
### Índices

- Foram criados índices para melhorar o desempenho das consultas, incluindo índices nas tabelas de dimensões para campos frequentemente consultados e

índices na tabela de Fato Notas para as chaves estrangeiras e outros campos relevantes.

## Modelagem Star Schema

- Com a criação das chaves estrangeiras já foi possível fazer a modelagem dimensional (Modelagem Star Schema) dentro do próprio PostgreSQL, e com isso ao carregar esses dados em uma ferramenta de análise de dados (Power BI por exemplo).



Modelagem Star Schema criada no PostgreSQL através da criação das chaves estrangeiras e carregadas no Power BI.

## Script SQL para criação do Data Warehouse:

-- Criação das tabelas de dimensões

-- Dimensão Candidato

```
CREATE TABLE dim_candidato (  
    CANDIDATO_ID SERIAL PRIMARY KEY,
```

```
NU_INSCRICAO BIGINT NOT NULL,  
FAIXA_ETARIA VARCHAR(100),  
SEXO CHAR(1),  
ETNIA VARCHAR(50),  
ESTADO_CIVIL VARCHAR(255),  
NACIONALIDADE VARCHAR(50),  
ST_CONCLUSAO VARCHAR(255),  
TREINEIRO VARCHAR(50),  
);
```

#### **-- Dimensão Localidade**

```
CREATE TABLE dim_localidade (  
    LOCALIDADE_ID SERIAL PRIMARY KEY,  
    UF VARCHAR(50),  
    NOME_MUNICIPIO VARCHAR(255)  
);
```

#### **-- Dimensão Prova**

```
CREATE TABLE dim_presenca (  
    PRESENCA_ID SERIAL PRIMARY KEY,  
    PRESENCA_CN VARCHAR(255),  
    PRESENCA_CH VARCHAR(255),  
    PRESENCA_LC VARCHAR(255),  
    PRESENCA_MT VARCHAR(255)  
);
```

#### **-- Dimensão Escola**

```
CREATE TABLE dim_escola (  
    ESCOLA_ID SERIAL PRIMARY KEY,  
    TIPO_ESCOLA VARCHAR(255),
```

```
LOCALIZACAO_ESC VARCHAR(255)
);
```

#### **-- Dimensão Tempo**

```
Create table dim_tempo (
    id_tempo SERIAL PRIMARY KEY,
    ano date
)
```

#### **-- Tabela de Fato Notas**

```
CREATE TABLE fat_notas (
    FATO_ID SERIAL PRIMARY KEY,
    CANDIDATO_ID INT,
    ESCOLA_ID INT,
    PRESENCA_ID INT,
    LOCALIDADE_ID INT,
    NOTA_PCN DECIMAL(7,2),
    NOTA_PCH DECIMAL(7,2),
    NOTA_PLC DECIMAL(7,2),
    NOTA_PMT DECIMAL(7,2),
    NOTA_REDACAO DECIMAL(7,2),
    NOTA_COMP1 DECIMAL(7,2),
    NOTA_COMP2 DECIMAL(7,2),
    NOTA_COMP3 DECIMAL(7,2),
    NOTA_COMP4 DECIMAL(7,2),
    NOTA_COMP5 DECIMAL(7,2)
);
```

#### **-- Adição de Restrições de Chave Estrangeira (FKs)**

```
ALTER TABLE fat_notas
```

```
ADD CONSTRAINT fk_candidato_id FOREIGN KEY (CANDIDATO_ID)
REFERENCES dim_candidato(CANDIDATO_ID);
```

```
ALTER TABLE fat_notas
```

```
ADD CONSTRAINT fk_escola_id FOREIGN KEY (ESCOLA_ID) REFERENCES
dim_escola(ESCOLA_ID);
```

```
ALTER TABLE fat_notas
```

```
ADD CONSTRAINT fk_presenca_id FOREIGN KEY (PRESENCA_ID)
REFERENCES dim_presenca(PRESENCA_ID);
```

```
ALTER TABLE fat_notas
```

```
ADD CONSTRAINT fk_localidade_id FOREIGN KEY (LOCALIDADE_ID)
REFERENCES dim_localidade(LOCALIDADE_ID);
```

```
ALTER TABLE fat_notas
```

```
ADD CONSTRAINT FK_fat_notas_ano FOREIGN KEY (TEMPO_ID)
REFERENCES dim_tempo(id_tempo);
```

### **-- Criação de Índices (opcional, para performance)**

```
CREATE INDEX idx_dim_candidato_inscricao ON dim_candidato(NU_INSCRICAO);
```

```
CREATE INDEX idx_dim_localidade_estado_municipio ON dim_localidade(UF,
NOME_MUNICIPIO);
```

```
CREATE INDEX idx_dim_escola_tipo_escola ON dim_escola(TIPO_ESCOLA);
```

### **-- Índices na tabela de fatos**

```
CREATE INDEX idx_fat_notas_candidato_id ON fat_notas(CANDIDATO_ID);
```

```
CREATE INDEX idx_fat_notas_escola_id ON fat_notas(ESCOLA_ID);
```

```
CREATE INDEX idx_fat_notas_localidade_id ON fat_notas(LOCALIDADE_ID);
```