

January 10, 2025

1 1. Trabalho Prático 2

2 2. Membros

- **Luís Eduardo Limas Brito:** 2023102329
- **Felipe Araujo Melo:** 2023027947
- **Natanael dos Santos Júnior:** 2023087842
- **Ulisses Drumond Souza Rosa:** 2023028161

3 3. Descrição dos dados

- **URL:** <https://dados.gov.br/dados/conjuntos-dados/dados-consolidados-de-movimentacao-de-gas-natural-em-gasodutos-de-transporte>
- Nesse trabalho, usamos os dados de *Julho de 2021*

Os dados escolhido armazenam as informações de movimentações de recursos entre diversas entidades. Analisando superficialmente os dados e seus identificadores, identificamos a existência das seguintes entidades: Instalação de Transporte, Instalação de Gasoduto, Município, Operador de Instalação e Carregador de Instalação. As movimentações entre as entidades citadas ocorrem por meio de um contrato, o qual retrata os eventos dessa movimentação separado por tipos de análises e separado por dia.

Os dados foram processados por linha. Pegamos os valores das colunas de interesse e adicionamos as tabelas correspondentes. Quanto aos dados, com exceção dos códigos de identificação e o valor do contrato, todos foram considerados do tipo **TEXTO**, enquanto o valor do contrato é do tipo **FLOAT** e os códigos são do tipo **INTEIRO**. Houve um problema na hora de implementação quanto ao código de identificação da entidade Carregador, pois um dos códigos era grande demais, e seu valor era traduzido para “2,75E+13”, o qual não era possível traduzir para **INTEIRO**. Por isso, decidimos que o código do Carregador será do tipo **TEXTO**.

A leitura dos dados ocorrerá em uma das células mais abaixo, porém antes precisamos criar as tabelas. Assim teremos que explicar a criação do **Modelo Relacional** e **Modelo ER** antes.

3.1 Inicialização do Programa

```
[1]: import io
import sqlite3
import pandas as pd
import numpy
import csv
from IPython.display import clear_output

print('Importacao concluida')
```

Importacao concluida

3.2 Conexão com o arquivo csv

```
[2]: #dataset: https://dados.gov.br/dados/conjuntos-dados/
    ↪ dados-consolidados-de-movimentacao-de-gas-natural-em-gasodutos-de-transporte
#data dos dados: julho de 2021

conn = sqlite3.connect('./Movimentacao_Julho_2021.db')
cursor = conn.cursor()

print('Setup concluido')
```

Setup concluido

4 4. Modelo Relacional

Para montar as tabelas, precisamos normalizar os dados. Isso quer dizer quebrar a linha dos dados em tabelas condizentes com as colunas apresentadas.

Os dados estão no formato:

- **Transportadora** (Código da Instalação de Transporte, Nome da Instalação de Transporte, Código da Instalação de Gasoduto, Nome da Instalação de Gasoduto, Tipo de Instalação de Gasoduto, Nome do Município da Instalação de Gasoduto, Nome da UF da Instalação de Gasoduto, Código do Operador da Instalação de Gasoduto, Nome do Operador da Instalação de Gasoduto, Código do Carregador que usa a Instalação de Gasoduto, Nome do Carregador que usa a Instalação de Gasoduto, Nome do Contrato da Instalação de Gasoduto, Nome da Variável, (Data, Valor));

A Primeira forma normal (1FN) não pode conter valores multivalorados. Assim, separamos eles em sua própria tabela:

- **Transportadora** (Código da Instalação de Transporte, Nome da Instalação de Transporte, Código da Instalação de Gasoduto, Nome da Instalação de Gasoduto, Tipo de Instalação de Gasoduto, Nome do Município da Instalação de Gasoduto, Nome da UF da Instalação de Gasoduto, Código do Operador da Instalação de Gasoduto, Nome do Operador da Instalação de Gasoduto, Código do Carregador que usa a Instalação de Gasoduto, Nome do Carregador

que usa a Instalação de Gasoduto, Nome do Contrato da Instalação de Gasoduto, Nome da Variável);

- **Transportadora_Contrato** (Código da Instalação de Transporte, Código da Instalação de Gasoduto, Nome do Município da Instalação de Gasoduto, Código do Operador da Instalação de Gasoduto, Código do Carregador que usa a Instalação de Gasoduto, Nome do Contrato da Instalação de Gasoduto, Nome da Variável, Data, Valor);

A Segunda forma normal (2FN) separa os atributos em tabelas onde a chave contém atributos que dependem dela completamente, e nada mais. Dessa forma, identificamos as seguintes dependências:

- **Código da Instalação de Transporte** → Nome da Instalação de Transporte, Código da Instalação de Gasoduto;
- **Código da Instalação de Gasoduto** → Nome da Instalação de Gasoduto, Tipo de Instalação de Gasoduto;
- **Nome do Município da Instalação de Gasoduto, Nome da UF da Instalação de Gasoduto** → Código da Instalação de Gasoduto;
- **Código do Operador da Instalação de Gasoduto** → Nome do Operador da Instalação de Gasoduto;
- **Código do Carregador que usa a Instalação de Gasoduto** → Nome do Carregador que usa a Instalação de Gasoduto;
- **Código da Instalação de Gasoduto, Código do Operador da Instalação de Gasoduto, Código do Carregador que usa a Instalação de Gasoduto** → Nome do Contrato da Instalação de Gasoduto;
- **Nome do Contrato da Instalação de Gasoduto, Nome da Variável, Código da Instalação de Gasoduto, Data** → Valor;

Adiantando a Terceira forma normal (3FN), temos uma dependência transitiva, que ocorre com o Nome do Contrato da Instalação do Gasoduto, que depende da instalação, do operador e do carregador e também tem suas dependências. Por isso, separamos em duas tabelas, uma relacionando o Nome do Contrato com a instalação, operador e carregador, e outra juntando os atributos do Contrato em si.

Outra dependência transitiva ocorre no Código do Gasoduto, que só existe no local do município, além de depender do transporte. Para tal, separamos ele em uma tabela de relação, que junta a Instalação do Gasoduto de seu município e outra que junta a Instalação de Transporte com a Instalação do Gasoduto; Logo, separando cada dependência em sua própria tabela temos as seguintes tabelas:

- **Instalação de Transporte** (Código da Instalação de Transporte, Nome da Instalação de Transporte);
- **Instalação de Gasoduto** (Código da Instalação de Gasoduto, Nome da Instalação de Gasoduto, Tipo de Instalação de Gasoduto);
- **Transporte_Gasoduto** (Código da Instalação de Gasoduto, Código da Instalação de Transporte);
- **Município** (Nome do Município da Instalação de Gasoduto, Nome da UF da Instalação de Gasoduto);
- **Município_Gasoduto** (Código da Instalação de Gasoduto, Nome do Município da Instalação do Gasoduto, Nome da UF da Instalação do Gasoduto);
- **Operador** (Código do Operador da Instalação de Gasoduto, Nome do Operador da Instalação de Gasoduto);

- **Carregador** (Código do Carregador que usa a Instalação de Gasoduto, Nome do Carregador que usa a Instalação de Gasoduto);
- **Instalação_Contrato** (Código da Instalação de Gasoduto, Código do Operador da Instalação de Gasoduto, Código do Carregador que usa a Instalação de Gasoduto, Nome do Contrato da Instalação de Gasoduto);
- **Contrato** (Nome do Contrato da Instalação de Gasoduto, Nome da Variável, Código da Instalação de Gasoduto, Data, Valor);

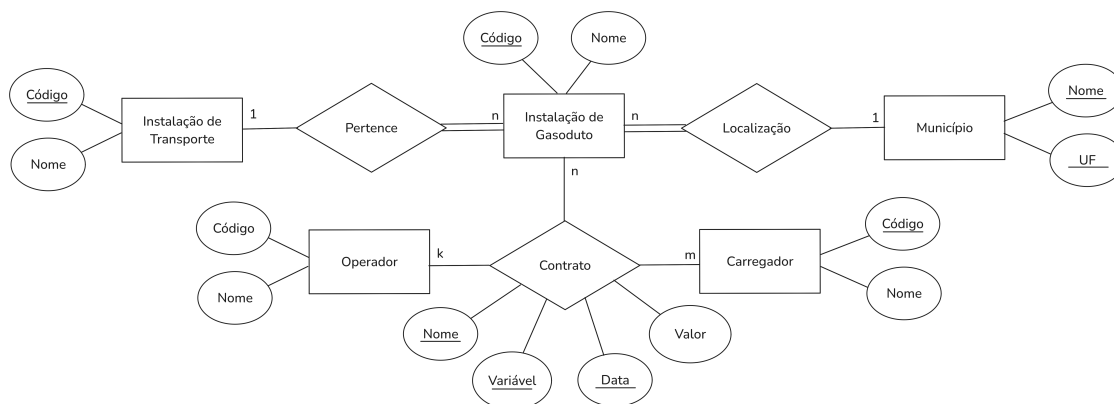
A fim de facilitar a modelação do banco de dados, decidimos simplificar os nomes, uma vez que não haverá conflito depois da separação de tabelas;

- **Instalação_de_Transporte** (Código, Nome);
- **Instalação_de_Gasoduto** (Código, Nome, Tipo);
- **Transporte_Gasoduto** (Código_do_Gasoduto, Código_do_Transporte);
 - Código_do_Transporte REFERENCIA Instalação_de_Trnsporte.Código;
- **Município** (Nome, UF);
- **Município_Gasoduto** (Código_do_Gasoduto, Nome_do_Município, UF_do_Município);
 - Nome_do_Município REFERENCIA Município.Nome;
 - UF_do_Município REFERENCIA Município.UF;
- **Operador** (Código, Nome);
- **Carregador** (Código, Nome);
- **Instalação_Contrato** (Código_do_Gasoduto, Código_do_Operador, Código_do_Carregador, Nome_do_Contrato);
 - Nome_do_Contrato REFERENCIA Contrato.Nome;
- **Contrato** (Nome, Variável, Código_do_Gasoduto, Data, Valor);
 - Código_do_Gasoduto REFERENCIA Instalação_de_Gasoduto.Código;

5 5. Modelo ER

O Modelo ER foi desenvolvido baseado no Modelo Relacional, porém com leves alterações. As tabelas Transporte_Gasoduto e Município_Gasoduto foram transformadas em relações do tipo 1:N, uma vez que cada instalação de gasoduto possui um município e uma instalação de transporte apenas, mas ambos o município e a instalação de transporte estão ligadas a mais de uma instalação de gasoduto.

Em especial, o Contrato se tornou uma gigante relação, envolvendo a Instalação de Gasoduto, o Operador da instalação e o Carregador da instalação. Decidimos por transformar o Contrato em uma relação pois o contrato em si depende totalmente das três entidades acima, tanto que seu nome é atrelado a eles.



Finalmente, montado o diagrama, criaremos as tabelas:

5.1 Criação das Tabelas

Antes de montar as tabelas, temos que solucionar as relações. Nas duas relações do tipo $1:N$ (**Pertence** e **Localização**), temos que cada **Instalação de Gasoduto** pode aparecer apenas uma vez, enquanto a outra entidade pode aparecer várias vezes. Nesse caso, mostrou-se vantajoso adicionar um atributo para cada relação (duas no total) que referenciem as entidades correspondente. Essas variáveis serão armazenadas na **tabela da Instalação de Gasoduto**.

A outra relação envolve três entidades (**Instalação de Gasoduto**, **Operador** e **Carregador**), sendo que cada uma não possui limite de vezes em que pode aparecer. Para tal, a única solução plausível era a criação de uma tabela que agrupe um referenciador para cada entidade envolvida, além dos atributos da própria relação. Essa tabela foi denominada **Contrato**. * Note que instâncias diferentes do Contrato pode envolver as mesmas entidades, por isso, Nome, Variável e Data também são chaves, pois elas afetam diretamente qual foi o valor do evento.

- **Importante:** Foi adicionado um novo atributo para Município, **codigo**, para garantir que cada município é único, além de facilitar o referenciamento por outras entidades. Dessa forma, **codigo** será usado como chave da tabela e não mais **nome** e **UF**

```
[3]: table_Instalacao_de_Transporte = '''CREATE TABLE IF NOT EXISTS_
↳Instalacao_de_Transporte(
    codigo INT,
    nome TEXT,
    PRIMARY KEY (codigo)
);'''
cursor.execute(table_Instalacao_de_Transporte)

table_Municipio = '''CREATE TABLE IF NOT EXISTS Municipio(
    codigo INT,
    nome TEXT,
    UF TEXT,
    PRIMARY KEY (codigo)
);'''
```

```

cursor.execute(table_Municipio)

table_Instalacao_de_Gasoduto = '''CREATE TABLE IF NOT EXISTS_
↳Instalacao_de_Gasoduto(
    codigo INT,
    nome TEXT,
    tipo TEXT,
    codigo_transporte INT,
    codigo_municipio INT,
    PRIMARY KEY (codigo),
    FOREIGN KEY (codigo_transporte)
        REFERENCES Instalacao_de_Transporte(codigo),
    FOREIGN KEY (codigo_municipio)
        REFERENCES Municipio(codigo)
);'''
cursor.execute(table_Instalacao_de_Gasoduto)

table_Operador = '''CREATE TABLE IF NOT EXISTS Operador(
    codigo INT,
    nome TEXT,
    PRIMARY KEY (codigo)
);'''
cursor.execute(table_Operador)

table_Carregador = '''CREATE TABLE IF NOT EXISTS Carregador(
    codigo TEXT,
    nome TEXT,
    PRIMARY KEY (codigo)
);'''
cursor.execute(table_Carregador)

table_Contrato = '''CREATE TABLE IF NOT EXISTS Contrato(
    nome TEXT,
    variavel TEXT,
    data TEXT,
    valor FLOAT,
    codigo_gasoduto INT,
    codigo_operador INT,
    codigo_carregador TEXT,
    PRIMARY KEY (codigo_gasoduto, codigo_operador, codigo_carregador, nome,
↳variavel, data),
    FOREIGN KEY (codigo_gasoduto)
        REFERENCES Instalacao_de_Gasoduto(codigo),

```

```

        FOREIGN KEY (codigo_operador)
            REFERENCES Operador(codigo),
        FOREIGN KEY (codigo_carregador)
            REFERENCES Carregador(codigo)
    );'''
cursor.execute(table_Contrato)

#delete = '''DROP TABLE Municipio'''
#cursor.execute(delete)
#delete = '''DROP TABLE Instalacao_de_Gasoduto'''
#cursor.execute(delete)

conn.commit()

print('Criacao concluida')

```

Criacao concluida

5.2 Leitura do Arquivo CSV

```

[4]: file = pd.read_csv('./gn-julho-2021.csv', chunksize=1, encoding='latin1',
    ↪ delimiter=';')
for line in file:

    # Impressai
    clear_output(wait=True)
    print('Fazendo leitura: ')
    print(int((line.index.start + 1)/2129 * 100), '% (', line.index.start + 1,
    ↪ '/ 2129 )')

    # Instalacao_Transporte
    transporte = line[['Código da Instalação de Transporte', 'Nome da
    ↪ Instalação de Transporte']].rename(columns={'Código da Instalação de
    ↪ Transporte':'codigo', 'Nome da Instalação de Transporte':'nome'})
    cursor.execute('''INSERT OR IGNORE INTO Instalacao_de_Transporte(codigo,
    ↪ nome)
        VALUES (?,?);''', (int(transporte['codigo'].values[0]), transporte['nome'].
    ↪ values[0]))

    # Municipio
    municipio = line[['Nome do Município da Instalação de Gasoduto', 'Nome da
    ↪ UF da Instalação de Gasoduto']].rename(columns={'Nome do Município da
    ↪ Instalação de Gasoduto':'nome', 'Nome da UF da Instalação de Gasoduto':'UF'})
    query = ('''SELECT * FROM Municipio as M WHERE M.nome = "name" AND M.UF =
    ↪ "uf"''').replace('name', str(municipio['nome'].values[0])).replace('uf',
    ↪ str(municipio['UF'].values[0]))
    if (pd.read_sql_query(query, conn).empty):

```

```

        cursor.execute('''INSERT OR IGNORE INTO Municipio(codigo, nome, UF)
        VALUES (?, ?, ?);''', (line.index.start + 1, municipio['nome'].values[0],
        ↳municipio['UF'].values[0]))

    # Instalacao de Gasoduto
    gasoduto = line[['Código da Instalação de Gasoduto', 'Nome da Instalação de
    ↳Gasoduto', 'Tipo da Instalação do Gasoduto', 'Código da Instalação de
    ↳Transporte', 'Nome do Município da Instalação de Gasoduto', 'Nome da UF da
    ↳Instalação de Gasoduto']].rename(columns=
        {'Código da Instalação de Gasoduto': 'codigo', 'Nome da Instalação de
    ↳Gasoduto': 'nome', 'Tipo da Instalação do Gasoduto': 'tipo', 'Código da
    ↳Instalação de Transporte': 'codigo_transporte', 'Nome do Município da
    ↳Instalação de Gasoduto': 'nome_municipio', 'Nome da UF da Instalação de
    ↳Gasoduto': 'UF_municipio'})
    mun = pd.read_sql_query('''SELECT * FROM Municipio as M WHERE M.nome =
    ↳"name" AND M.UF = "uf"'''.replace('name', str(municipio['nome'].values[0])).
    ↳replace('uf', str(municipio['UF'].values[0])), conn)
    if (not mun.empty):
        cursor.execute('''INSERT OR IGNORE INTO Instalacao_de_Gasoduto(codigo,
    ↳nome, tipo, codigo_transporte, codigo_municipio)
        VALUES (?, ?, ?, ?, ?);''', (float(gasoduto['codigo'].values[0]),
    ↳gasoduto['nome'].values[0], gasoduto['tipo'].values[0],
    ↳int(gasoduto['codigo_transporte'].values[0]), int(mun['codigo'].values[0])))

    # Operador
    operador = line[['Código do Operador da Instalação de Gasoduto', 'Nome do
    ↳Operador da Instalação de Gasoduto']].rename(columns={'Código do Operador da
    ↳Instalação de Gasoduto': 'codigo', 'Nome do Operador da Instalação de
    ↳Gasoduto': 'nome'})
    cursor.execute('''INSERT OR IGNORE INTO Operador(codigo, nome)
    VALUES (?, ?);''', (int(operador['codigo'].values[0]), operador['nome'].
    ↳values[0]))

    # Carregador
    carregador = line[['Código do Carregador que usa a Instalação de Gasoduto',
    ↳'Nome do Carregador que usa a Instalação de Gasoduto']].
    ↳rename(columns={'Código do Carregador que usa a Instalação de Gasoduto':
    ↳'codigo', 'Nome do Carregador que usa a Instalação de Gasoduto': 'nome'})
    if (isinstance(carregador['codigo'].values[0], str)):
        cursor.execute('''INSERT OR IGNORE INTO Carregador(codigo, nome)
        VALUES (?, ?);''', (carregador['codigo'].values[0], carregador['nome'].
    ↳values[0]))
    else:
        cursor.execute('''INSERT OR IGNORE INTO Carregador(codigo, nome)

```



```

VALUES (?,?);''' , (float(carregador['codigo'].values[0]),
↪carregador['nome'].values[0]))

# Contrato
for i in range (13, 44): # Dias 1 - 31
    contrato = line[['Nome do Contrato da Instalação do Gasoduto', 'Nome da
↪variável', line.columns[i], 'Código da Instalação de Gasoduto', 'Código do
↪Operador da Instalação de Gasoduto', 'Código do Carregador que usa a
↪Instalação de Gasoduto']].rename(columns=
        {'Nome do Contrato da Instalação do Gasoduto':'nome', 'Nome da
↪variável':'variavel', line.columns[i]: 'data', 'Código da Instalação de
↪Gasoduto':'codigo_gasoduto', 'Código do Operador da Instalação de Gasoduto':
↪'codigo_operador',
        'Código do Carregador que usa a Instalação de Gasoduto':
↪'codigo_carregador'})
    if (isinstance(contrato['codigo_carregador'].values[0], str)):
        codigo_carregador = contrato['codigo_carregador'].values[0]
    else:
        codigo_carregador = float(contrato['codigo_carregador'].values[0])
    if (contrato['variavel'].values[0] == 'Alocação'):
        data_valor = numpy.float64((str(int(contrato['data'].values[0])) +
↪',00').replace(',','.')))
    else:
        data_valor = numpy.float64((str(contrato['data'].values[0])).
↪replace(',','.')))
    cursor.execute('''INSERT OR IGNORE INTO Contrato(nome, variavel, data,
↪valor, codigo_gasoduto, codigo_operador, codigo_carregador)
        VALUES (?, ?, ?, ?, ?, ?, ?);''' , (contrato['nome'].values[0],
↪contrato['variavel'].values[0], line.columns[i], data_valor,
        float(contrato['codigo_gasoduto'].values[0]),
↪int(contrato['codigo_operador'].values[0]), codigo_carregador))
;

print("Deletando valores nulos")

delete = '''DELETE FROM Instalacao_de_Transporte WHERE codigo IS NULL;'''
cursor.execute(delete)
delete = '''DELETE FROM Municipio WHERE nome IS NULL;'''
cursor.execute(delete)
delete = '''DELETE FROM Instalacao_de_Gasoduto WHERE codigo IS NULL;'''
cursor.execute(delete)
delete = '''DELETE FROM Operador WHERE codigo IS NULL;'''
cursor.execute(delete)
delete = '''DELETE FROM Carregador WHERE codigo IS NULL;'''
cursor.execute(delete)
delete = '''DELETE FROM Contrato WHERE valor IS NULL OR valor = 0.0;'''

```

```

cursor.execute(delete)

conn.commit()

print('Leitura Concluida')

```

Fazendo leitura:
 100 % (2129 / 2129)
 Deletando valores nulos
 Leitura Concluida

5.3 Leitura do Banco de Dados

Caso as tabelas já foram criadas e a leitura já foi feita, **pule para aqui** para não precisar refazer os passos anteriores

```

[5]: conn = sqlite3.connect('./Movimentacao_Julho_2021.db')
      cursor = conn.cursor()

      print('Banco de Dados Conectado')

```

Banco de Dados Conectado

6 6. Consultas

6.1 6.1 Duas consultas envolvendo seleção e projeção

6.1.1 6.1.1 Consulta 1

Instalação de Gasodutos do tipo Recebedor

Essa pesquisa encontra as instalações capazes de receber os recursos. Útil para encontrar possibilidades de lugares para enviar ditos recursos.

```

[6]: query = '''
      SELECT DISTINCT
          G.codigo as Codigo, G.nome as Nome
      FROM
          Instalacao_de_Gasoduto as G
      WHERE
          G.tipo = "Ponto de Recebimento"
      ORDER BY
          G.codigo;
      '''

      pr = pd.read_sql_query(query, conn)
      pr

```

[6]:	Codigo	Nome
0	18368	Corumbá (Mutun)
1	31083	Interconexão Ipojuca (Pilar-Ipojuca >> EDG Ipo...
2	31262	Guamaré II (Nordestão)
3	53684	Interconexão REPLAN (GASBOL >> GASCAR)
4	111505	Interconexão Paulínia (GASBOL >> GASPAJ)
5	111567	Cáceres
6	212585	Uruguaiana
7	212587	Canoas
8	217953	Atalaia II (UPGN Atalaia >> GAL)
9	217956	Interconexão Pilar (Pilar-Ipojuca >> EDG Pilar)
10	217962	Pojuca II (UPGN Santiago)
11	217963	Interconexão Catu (GASCAC >> EDG Catu)
12	217980	São Francisco do Conde I (UPGN Candeias)
13	217981	São Francisco do Conde II (EVF - Manati)
14	217982	São Francisco do Conde III (GNL TRBA)
15	219711	Guamaré I (GASFOR)
16	219843	GNL Pecém
17	220776	Interconexão Catu (EDG Catu >> GASCAC)
18	220777	São Sebastião do Passé (GNL TRBA)
19	220939	Cacimbas (UTGC)
20	220981	Interconexão TECAB (TECAB >> GASCAV)
21	220986	UTGSUL
22	221007	Interconexão Taubaté (GASTAU >> EDG Taubaté)
23	221210	Caraguatatuba (UTGCA)
24	221336	REDUC I
25	221337	REDUC II
26	221540	Interconexão Campos Elíseos I (GASDUC III >> E...
27	221720	GNL BG
28	221812	Interconexão Ipojuca (EDG Ipojuca >> Pilar-Ipo...
29	221875	Interconexão Volta Redonda (GASCAR >> GASBEL II)
30	221892	Interconexão Japeri (EDG Japeri >> GASJAP)
31	221893	Interconexão Campos Elíseos II (EDG Campos Elí...
32	222296	Interconexão TECAB (TECAB >> GASDUC III)
33	222297	Interconexão Campos Elíseos I (EDG Campos Elís...
34	222447	Interconexão Guararema I (GASBOL >> GASPAL)
35	222448	Interconexão ECGM (GASPAL II >> GASPAL)
36	222449	Interconexão São José dos Campos (GASTAU >> GA...
37	222743	Interconexão Guararema II (GASPAL >> GASPAL II)
38	222745	RPBC
39	222746	Interconexão ESBC (GASAN II >> GASAN)
40	271705	Guararema (EMR Guararema)
41	271706	GASCAR (EMR REPLAN)
42	285832	Interconexão Campos Elíseos II (GASJAP >> EDG ...
43	286017	Interconexão Japeri (GASJAP >> EDG Japeri)
44	286099	Interconexão Queluzito (GASBEL II >> GASBEL)
45	292241	Urucu (Pólo de Processamento de Arara)

```

46 292387          Interconexão SDV-02 (GASENE >> Lagoa Parda)
47 292582          Marechal Deodoro (UPGN Alagoas)
48 292597  Interconexão Pilar (EDG Pilar >> Pilar-Ipojuca)
49 292757          Atalaia I (UPGN Atalaia >> GAI)

```

6.1.2 6.1.2 Consulta 2

Contratos que geraram eventos no dia 30 de Julho

Essa pesquisa encontra os contratos por trás dos eventos do dia 30. Útil para descobrir os responsáveis das movimentações do dia.

```

[6]: query = '''
SELECT DISTINCT
    C.nome
FROM
    Contrato as C
WHERE
    C.data = "30/07/2021" AND C.nome IS NOT NULL
ORDER BY
    C.nome;
'''

pr = pd.read_sql_query(query, conn)
pr

```

```

[6]:
0                               nome
1                               Bilac
2          CP001CBA-E2021-PTB
3          CP001SC1-S2021-PTB
4          CP001SC2-S2021-PTB
5          CP0022GSC-E2021-PET
6          CP0023MS1-S2021-PET
7          CP0024SP1-S2021-PET
8          CP0025SP2-S2021-PET
9          CP0026SP4-S2021-PET
10                             CPAC 07
11      Canoas - Pólo Petroquímico (Trecho 3)
12      Fronteira (Argentina) - Uruguaiana (Trecho 1)
13                             GASDUC III
14          GASENE - Trecho Norte
15          GASENE - Trecho Sul
16                             GASTAU
17      Gasocidente do Mato Grosso Ltda x AMBAR Energi...
18      Gasocidente do Mato Grosso Ltda x Companhia Ma...
19                             Guaiçara
20                             Malha Nordeste
21                             Malhas II

```

21	Malhas SE
22	Paulínia-Jacutinga
23	Pilar-Ipojuca
24	Proquigel Entrada
25	Proquigel Saída
26	TCO
27	TCX
28	Urucu-Coari-Manaus

6.2 6.2 Três consultas envolvendo junção de duas relações

6.2.1 6.2.1 Consulta 3

Instalações de Transporte em São Paulo - Para essa consulta, temos que unir **Instalacao_de_Transporte** com **Instalacao_de_Gasoduto** (Relação “*Pertence*”) e também com **Municipio** (Relação “*Localização*”).

Essa pesquisa encontra as Instalações de Transporte que operam em São Paulo. Essa pesquisa é interessante para o governo, que busca compreender as atuações dessas empresas de movimentação dentro de seu território

```
[7]: query = '''
SELECT DISTINCT
    T.codigo as Codigo, T.nome as Nome
FROM
    Instalacao_de_Transporte as T JOIN (SELECT *
    FROM Instalacao_de_Gasoduto as G JOIN Municipio as M
    ON G.codigo_municipio = M.codigo
    WHERE M.UF = "SP") as GM
ON
    T.codigo = GM.codigo_transporte
WHERE
    GM.UF = "SP";
'''

pr = pd.read_sql_query(query, conn)
pr
```

```
[7]:
```

	Codigo	Nome
0	700525	Paulínia-Jacutinga
1	700540	GASTAU
2	514070	Taubaté
3	700550	GASAN
4	700505	GASCAR
5	700544	GASPAL
6	700549	GASAN II
7	700548	GASPAL II
8	700503	GASBOL

6.2.2 6.2.2 Consulta 4

Contratos da Pilar - Para essa consulta, temos que unir **Instalacao_de_Transporte** com **Instalacao_de_Gasoduto** (Relação “*Pertence*”) e também com **Contrato** (Relação “*Contrato*”).

Essa pesquisa encontra todos os contratos nos quais a Pilar (Intalação de Transporte) estava envolvido. Útil para traçar as movimentações e eventos relacionados com a Pilar

```
[8]: query = '''
SELECT DISTINCT
    GC.nome as Nome
FROM
    Instalacao_de_Transporte as T JOIN (SELECT C.nome as nome, G.
↪codigo_transporte
    FROM Instalacao_de_Gasoduto as G JOIN Contrato as C
    ON C.codigo_gasoduto = G.codigo
    WHERE C.nome IS NOT NULL) GC
ON
    T.codigo = GC.codigo_transporte
WHERE
    T.nome = "GASBOL"
ORDER BY
    GC.nome;
'''

pr = pd.read_sql_query(query, conn)
pr
```

```
[8]:
```

	Nome
0	Bilac
1	CP001CBA-E2021-PTB
2	CP001SC1-S2021-PTB
3	CP001SC2-S2021-PTB
4	CP0022GSC-E2021-PET
5	CP0023MS1-S2021-PET
6	CP0024SP1-S2021-PET
7	CP0025SP2-S2021-PET
8	CP0026SP4-S2021-PET
9	CPAC 07
10	Guaíçara
11	TCO
12	TCX

6.2.3 6.2.3 Consulta 5

Municípios onde a Petrobrás não operou - Para essa consulta, temos que unir **Carregador** com **Instalacao_de_Gasoduto** (Relação “*Contrato*”) e também com **Município** (Relação “*Localização*”).

Essa pesquisa encontra todos os municípios onde a Petrobrás não fez carregamento. Essa pesquisa

serve, por exemplo, para empresas competidoras da Petrobrás que gostariam de se estabelecer em cidades onde a Petrobrás não opera.

```
[10]: query = '''
SELECT DISTINCT
    M.nome as Nome, M.UF as UF
FROM
    Municipio as M LEFT JOIN (SELECT DISTINCT G.codigo_municipio as_
↪codigo_municipio
    FROM Instalacao_de_Gasoduto as G JOIN ( SELECT DISTINCT Co.codigo_gasoduto_
↪as codigo_gasoduto
        FROM Contrato as Co JOIN Carregador as Ca
        ON Co.codigo_carregador = Ca.codigo
        WHERE Ca.nome = "Petróleo Brasileiro S.A. - PETROBRAS") as C
    ON C.codigo_gasoduto = G.codigo) as GC
ON
    M.codigo = GC.codigo_municipio
WHERE
    GC.codigo_municipio IS NULL
ORDER BY
    M.nome;
'''

pr = pd.read_sql_query(query, conn)
pr
```

```
[10]:      Nome  UF
0    Cuiabá  MT
1    Cáceres  MT
2    Triunfo  RS
3  Uruguaiana  RS
```

6.3 6.3 Três consultas envolvendo junção de três ou mais relações

Para os seguintes casos, tivemos que utilizar todas as três relações disponíveis (**Pertence**, **Localização** e **Contrato**).

6.3.1 6.3.1 Consulta 6

Instalações de Transporte que participaram de Solicitações de Volume no Rio de Janeiro no dia 15 de Julho.

Essa pesquisa serve para encontrar as Instalações de Transporte cujas Instalações de Gasoduto fizeram solicitações de recurso. Útil para verificação quanto foi pedido para verificação posterior de validação dos contratos

```
[11]: query = '''
SELECT DISTINCT
    T.codigo as Codigo, T.nome as Nome, SUM(MCG.valor) as Total_Valor
```

```

FROM
    Instalacao_de_Transporte as T JOIN
    (SELECT DISTINCT *
     FROM Municipio as M JOIN
        (SELECT DISTINCT *
         FROM Contrato as C JOIN Instalacao_de_Gasoduto as G
          ON C.codigo_gasoduto = G.codigo
          WHERE C.variavel = "Volume Solicitado" AND C.data = "15/07/2021") as CG
     ON CG.codigo_municipio = M.codigo
     WHERE M.UF = "RJ") as MCG
ON
    T.codigo = MCG.codigo_transporte
GROUP BY
    T.codigo
ORDER BY
    SUM(MCG.valor) DESC;
'''

pr = pd.read_sql_query(query, conn)
pr

```

```

[11]:
   Codigo      Nome  Total_Valor
0  514265  Campos Elíseos    40975.47
1  700543    GASDUC III    27217.66
2  700542    GASJAP     19462.00
3  700505    GASCAR    12638.96
4  514180   Anel de Gás     7568.67
5  700545    GASBEL II     3721.00
6  700539    GASCAV     3658.69
7  700506    GASVOL     2094.72
8  700544    GASPAL      409.82
9  700541    GASBEL       55.03

```

6.3.2 6.3.2 Consulta 7

Instalação de Transporte que operam na Região Sul com maior quantidade de Contratos no dia 7 de Julho.

Essa pesquisa encontra as Instalações de Transporte operando na Região Sul cujos contratos levaram a eventos o dia 7 de Julho, junto da quantidade desses contratos por Transporte. Útil para descobrir as Instalações de Transporte que tiveram movimentação nesse dia.

```

[12]: query = '''
SELECT DISTINCT
    T.codigo as Codigo, T.nome as Nome, COUNT(MCG.nome) as Num_Contratos
FROM
    Instalacao_de_Transporte as T JOIN
    (SELECT DISTINCT CG.nome as nome, CG.codigo_transporte as codigo_transporte

```



```

FROM Municipio as M JOIN
    (SELECT DISTINCT C.nome as nome, G.codigo_municipio as_
↪codigo_municipio, G.codigo_transporte as codigo_transporte
    FROM Contrato as C JOIN Instalacao_de_Gasoduto as G
    ON C.codigo_gasoduto = G.codigo
    WHERE C.data = "07/07/2021") as CG
ON CG.codigo_municipio = M.codigo
WHERE M.UF = "RS" OR M.UF = "SC" OR M.UF = "PR") as MCG
ON
    T.codigo = MCG.codigo_transporte
GROUP BY
    T.codigo
ORDER BY
    COUNT(MCG.nome) DESC;
'''

pr = pd.read_sql_query(query, conn)
pr

```

```

[12]:
   Codigo      Nome  Num_Contratos
0  700503      GASBOL             5
1  505445  Uruguaiana - Porto Alegre      1
2  505444  Uruguaiana - Porto Alegre      1

```

6.3.3 6.3.3 Consulta 8

Carregadores que já trabalharam para GASBEL e GASBEL II fora de Minas Gerais.

Essa pesquisa encontra os Carregadores que já fecharam contrato com a GASBEL E GASBEL II em um trabalho fora de Minas Gerais. Como GASBEL e GASBEL II atuam primariamente em Minas Gerias, julga-se interessante para a empresa olhar para fora e analisar novas oportunidades sem repetir lugares já passados.

```

[13]: query = '''
SELECT DISTINCT
    MCG.ca_codigo as Codigo, MCG.ca_nome as Nome, MCG.mun_nome as Municipio,
↪MCG.mun_UF as UF
FROM
    Instalacao_de_Transporte as T JOIN
    (SELECT DISTINCT CG.ca_codigo as ca_codigo, CG.ca_nome as ca_nome, CG.
↪codigo_transporte as codigo_transporte, M.nome as mun_nome, M.UF as mun_UF
    FROM Municipio as M JOIN
        (SELECT DISTINCT C.ca_codigo as ca_codigo, C.ca_nome as ca_nome, G.
↪codigo_transporte as codigo_transporte, G.codigo_municipio as_
↪codigo_municipio
        FROM Instalacao_de_Gasoduto as G JOIN (SELECT DISTINCT Ca.codigo as_
↪ca_codigo, Ca.nome as ca_nome, Co.codigo_gasoduto as codigo_gasoduto
        FROM Carregador as Ca JOIN Contrato as Co

```

```

        ON Ca.codigo = Co.codigo_carregador) C
        ON C.codigo_gasoduto = G.codigo) as CG
    ON CG.codigo_municipio = M.codigo
    WHERE M.UF != "MG") as MCG
ON
    T.codigo = MCG.codigo_transporte
WHERE
    T.nome = "GASBEL" OR T.nome = "GASBEL II"
ORDER BY
    MCG.ca_codigo DESC;
'''

pr = pd.read_sql_query(query, conn)
pr

```

```

[13]:
      Codigo      Nome      Municipio UF
0  33000167.0  Petróleo Brasileiro S.A. - PETROBRAS  Rio das Flores RJ
1  33000167.0  Petróleo Brasileiro S.A. - PETROBRAS  Volta Redonda RJ

```

6.4 Duas consultas envolvendo agregação sobre junção de duas ou mais relações

6.4.1 Consulta 9

Municípios com maior valor de Volume Realizado - Para essa consulta, temos que unir **Contrato** com **Instalacao_de_Gasoduto** (Relação “*Contrato*”) e também com **Municipio** (Relação “*Localização*”).

Essa pesquisa encontra os Municipios com maior volume de recursos utilizados. Ela é interessante para analisar municípios produtíveis quanto às suas operações.

```

[14]: query = '''
SELECT DISTINCT
    M.nome as Nome, M.UF as UF, SUM(GC.valor) as Valor_Acumulado
FROM
    Municipio as M JOIN (SELECT DISTINCT G.codigo_municipio as L
    ↪codigo_municipio, C.data as data, C.valor as Valor
    FROM Instalacao_de_Gasoduto as G JOIN Contrato as C
    ON C.codigo_gasoduto = G.codigo
    WHERE C.variavel = "Volume Realizado") as GC
ON
    M.codigo = GC.codigo_municipio
GROUP BY
    M.codigo
ORDER BY
    SUM(GC.valor) DESC;
'''

```

```
pr = pd.read_sql_query(query, conn)
pr
```

```
[14]:
```

	Nome	UF	Valor_Acumulado
0	Duque de Caxias	RJ	1536666.48
1	Macaé	RJ	770344.34
2	Japeri	RJ	706358.92
3	Corumbá	MS	677498.77
4	Guararema	SP	541433.45
..
125	Valparaíso	SP	134.86
126	Mamanguape	PB	124.66
127	Itapetininga	SP	97.01
128	Itirapina	SP	25.28
129	São Paulo	SP	2.92

```
[130 rows x 3 columns]
```

6.4.2 6.4.2 Consulta 10

Quantidade de Município em que cada Instalação de Transporte opera - Para essa consulta, temos que unir **Instalacao_de_Transporte** com **Instalacao_de_Gasoduto** (Relação “*Pertence*”) e também com **Municipio** (Relação “*Localização*”).

Essa pesquisa encontra a quantidade de municípios ligados as Instalações de Transporte. Útil para compará-las quanto ao seu alcance e influência nas regiões.

```
[15]: query = '''
SELECT DISTINCT
    T.codigo as Codigo, T.nome as Nome, COUNT(GM.codigo) as Num_Municipio
FROM
    Instalacao_de_Transporte as T JOIN (SELECT DISTINCT M.codigo as codigo, G.
    codigo_transporte as codigo_transporte
    FROM Instalacao_de_Gasoduto as G JOIN Municipio as M
    ON G.codigo_municipio = M.codigo) as GM
ON
    T.codigo = GM.codigo_transporte
GROUP BY
    T.codigo
ORDER BY
    COUNT (GM.codigo) DESC;
'''

pr = pd.read_sql_query(query, conn)
pr
```

```
[15]:
```

	Codigo	Nome	Num_Municipio
0	700503	GASBOL	41

1	700512	Nordestão	13
2	700544	GASPAL	10
3	700514	Catu-Pilar	10
4	700536	GASFOR	9
5	700513	Urucu-Manaus	8
6	700505	GASCAR	8
7	700541	GASBEL	7
8	700520	GASCAC	6
9	700539	GASCAV	5
10	700506	GASVOL	4
11	700543	GASDUC III	3
12	700540	GASTAU	3
13	700522	GASENE-Lagoa Parda-VB-10	3
14	700507	GASALP	3
15	700550	GASAN	2
16	700548	GASPAL II	2
17	700545	GASBEL II	2
18	700542	GASJAP	2
19	700525	Paulínia-Jacutinga	2
20	700521	Bolívia - Mato Grosso Lateral Cuiabá	2
21	700511	Pilar-Ipojuca	2
22	700504	Cacimbas-Vitória	2
23	505445	Uruguaiana - Porto Alegre	2
24	700549	GASAN II	1
25	700533	Candeias-Camaçari 12"	1
26	700516	Candeias-Aratu 12"	1
27	514265	Campos Elíseos	1
28	514180	Anel de Gás	1
29	514070	Taubaté	1
30	512818	Aratu	1
31	512574	Catu	1
32	512573	Camaçari	1
33	512572	Pilar	1
34	512566	Atalaia	1
35	512556	São Francisco do Conde	1
36	505444	Uruguaiana - Porto Alegre	1

7 7. Autoavaliação dos membros

7.0.1 Felipe Melo:

Nesse trabalho prático fiquei encarregado de lidar com 5 consultas e implementar a parte extra. Acredito que desempenhei um bom trabalho pois todos os membros gostaram do resultado final do trabalho com a implementação do streamlit.

7.0.2 Luís Eduardo:

A minha parte no trabalho foi ler os dados da base de dados e montar as tabelas. Esse trabalho foi um bocado complicado pois a base de dados estava um pouco “desorganizada”. Entretanto no fim foi possível montar um banco de dados seguindo os relacionamentos propostos no esquema ER. Gostei muito de ficar com essa tarefa, uma vez que foi possível botar em prática vários dos conceitos aprendidos na disciplina. Todo mundo no grupo gostou da minha parte.

7.0.3 Nathanael dos Santos:

Eu fiquei com a parte de fazer 5 consultas e montar a documentação. Explicar as consultas de forma sucinta se mostrou um desafio, uma vez que o código por trás era tão complicado quanto a consulta. Por isso a comunicação entre nós foi crucial para a realização dessa tarefa. Além disso, eu também tive que explicar as minúcias do nosso trabalho. Fiquei satisfeito com o meu desempenho.

7.0.4 Ulisses Rosa:

No trabalho em questão a minha parte foi escolher o banco de dados e fazer o diagrama ER junto com a normalização das tabelas. Montar o diagrama ER foi desafiador pois interpretar a base de dados que escolhemos foi uma tarefa bem difícil. Gostei bastante do resultado do meu trabalho e também do resultado do trabalho dos outros membros.

8 8 Streamlit

Criamos uma versão desse trabalho em formato de aplicação StreamLit, contendo as consultas já feitas e extras onde o usuário é capaz de escolher as informações relevantes para ele.

Link: <https://tp2-ibd-movimentacao-2021.streamlit.app/>