Fundamentos de BD (Márcio Victorino)

Partes do Projeto:

1. Introdução.
2. Modelo de dados Relacional.
3. O script SQL que gerou o banco de dados.
4. Apresentar o processo de ETL (Extract, Transform, Load) para importação dos

dados para o banco de dados.

1. Utilização de pelo menos uma View.
2. Utilização de pelo menos uma Procedure (com comandos condicionais).
3. Utilização de pelo menos um trigger (com comandos condicionais).
4. No mínimo 5 Consultas SQL (a complexidade da consulta será avaliada).

Data Limite para a Entrega da Parte Escrita: 02/06/2023

Alunos:

Marcelo Anselmo de Souza Filho

- Matrícula: **231109719**
- Email: **marcelofilho@mpf.mp.br**

Arivaldo Gonçalves de Freitas Junior

- Matrícula: **231109620**
- Email: **arivaldofreitas@correios.com.br**

Luciana Maria de Araujo Freitas

- Matrícula: **231109700**
- Email: **luciana@mpdft.mp.br**

---

1. Introdução

Sobre: Este estudo aborda a diferença salarial entre homens e mulheres na área de TI durante a pandemia. Ele também explora possíveis cenários para analisar a disparidade salarial e de desligamento entre gêneros na área de tecnologia, antes e após a pandemia. Utilizou-se dados a nível do indivíduo, de 2018 e 2019 (antes da pandamiea) e de 2020 e 2021 (durante a pandemia), obtidos da Relação Anual de Informações Sociais (Rais), que proporciona dados oficiais sobre o mercado de trabalho no Brasil.

Tecnologias utilizadas:

- BD: Mysql (docker)
- Linguagem: Python
- Dados: RAIS

- Ambiente de DEV: VsCode + Jupyter Notebook

1.1 Carregando Bibliotecas necessárias

In [ ]:
```
import mysql.connector
import warnings
import base64
from base64 import b64decode
from IPython.display import Image, display
import pandas as pd

# MERMAID: para visualização de Diagramas no Markdown
def mm(graph):
  graph = graph.replace("\n", " ")
  graphbytes = graph.encode("ascii")
  base64_bytes = base64.b64encode(graphbytes)
  base64_string = base64_bytes.decode("ascii")
  url_img = "https://mermaid.ink/img/" + base64_string
  img_file = Image(url=url_img)
  display(img_file)
  return url_img

# Ignorando Warnings do Python
warnings.filterwarnings("ignore")

# Conexão com o Banco de Dados
def get_cnx():
  return mysql.connector.connect(user='root', password='root', database='projfbd')
cnx = get_cnx()
```

2. Modelo de dados Relacional

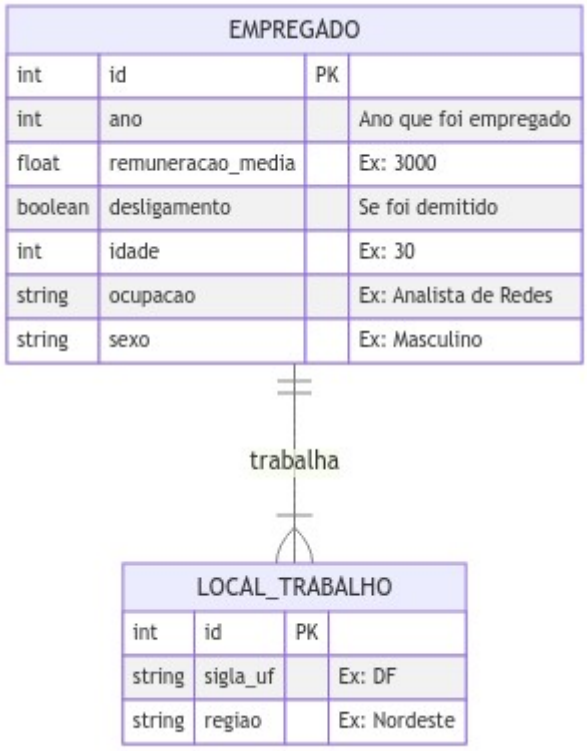
2.1 Modelo Conceitual

https://mermaid.js.org/syntax/entityRelationshipDiagram.html

Integrando jupyter com mermaid: https://mermaid.js.org/config/Tutorials.html#jupyter-integration-with-mermaid-js

In [ ]:
```
img_mm_1 = mm("""
erDiagram
    EMPREGADO {
        int id PK
        int ano "Ano que foi empregado"
        float remuneracao_media "Ex: 3000"
        boolean desligamento "Se foi demitido"
        int idade "Ex: 30"
        string ocupacao "Ex: Analista de Redes"
        string sexo "Ex: Masculino"
    }
    LOCAL_TRABALHO {
        int id PK
        string sigla_uf "Ex: DF"
        string regiao "Ex: Nordeste"
    }
```

```
    EMPREGADO ||--|{ LOCAL_TRABALHO : trabalha
""")
```



2.2 Modelo Lógico

2.2.1 Normalização

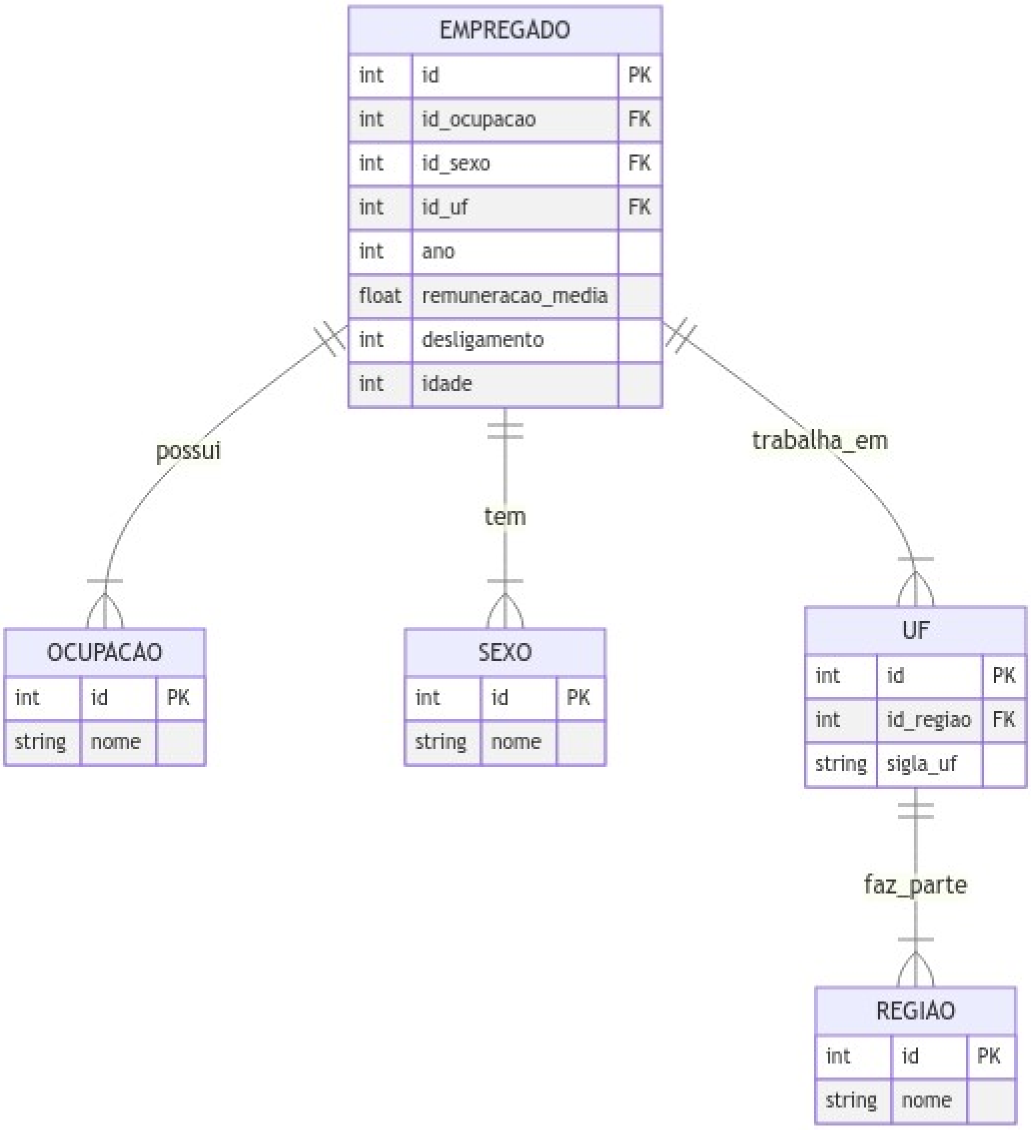
1º FN: Uma relação está em 1FN se e somente se todos os seus atributos contêm apenas valores atômicos (simples, indivisíveis)

RESPOSTA: A relação está na 1FN, pois todos os atributos são simples e indivisíveis.

2º FN: Uma relação encontra-se na 2FN se e somente se estiver em 1FN e não contém dependências parciais.

In [ ]:
```
img_mm_2 = mm("""
erDiagram
    EMPREGADO ||--|{ OCUPACAO : possui
    EMPREGADO ||--|{ SEXO : tem
    EMPREGADO ||--|{ UF : trabalha_em
    UF ||--|{ REGIAO : faz_parte
    EMPREGADO {
        int id PK
        int id_ocupacao FK
        int id_sexo FK
        int id_uf FK
        int ano
        float remuneracao_media
        int desligamento
        int idade
    }
    OCUPACAO {
        int id PK
        string nome
```

```
    }
    SEXO {
        int id PK
        string nome
    }
    UF {
        int id PK
        int id_regiao FK
        string sigla_uf
    }
    REGIAO {
        int id PK
        string nome
    }
""")
img_mm_2
```



Out[ ]: 'https://mermaid.ink/img/IGVyRGlhZ3JhbSAgICAgICAgIEVNUFJFR0FETyB8fC0tfHsgT0NVUEFDQU8gOiBwb3NzdWkgICAgIEVNUFJFR0FETyB8fC0tfHsgU0VYTyA6IHRlbSAgICAgRU1QUkVHQURPIHx8LS18eyBVRiA6IHRyYWJhbGhhX2VtICAgICBVRiB8fC0tfHsgUkVHSUFPIDogZmF6X3BhcnRlICAgICBFTVBSRUdBRE8geyAgICAgICAgIGludCBpZCBQSyAgICAgICAgIGludCBpZF9vY3VwYWNhbyBGSyAgICAgICAgIGludCBpZF9zZXhvIEZLICAgICAgICAgaW50IGlkX3VmIEZLICAgICAgICAgaW50IGFubyAgICAgICAgIGZsb2F0IHJlbXVuZXJhY2FvX21lZGlhICAgICAgICAgaW50IGRlc2xpZ2FtZW50byAgICAgICAgIGludCBpZGFkZSAgICAgfSAgICAgT0NVUEFDQU8geyAgICAgICAgIGludCBpZCBQSyAgICAgICAgIHN0cmluZyBub21lICAgICB9ICAgICBTRVhPIHsgICAgICAgIGludCBpZCBQSyAgICAgICAgIHN0cmluZyBub21lICAgICB9ICAgICBVRiB7ICAgICAgICAgaW50IGlkIFBLICAgICAgICAgaW50IGlkX3JlZ2lhbyBGSyAgICAgICAgIHN0cmluZyBzaWdsYV91ZiAgICAgfSAgICAgUkVHSUFPIHsgICAgICAgIGludCBpZCBQSyAgICAgICAgIHN0cmluZyBub21lICAgICB9IH0g'

3º FN: Uma relação está em 3FN se e somente se estiver na 2FN e nenhum atributo não-primo (isto é, que não seja membro de uma chave) for transitivamente dependente da chave primária.

RESPOSTA: A relação está na 3FN, pois não há dependência transitiva.

2.2.2 Especificação do BD

- OCUPACAO (id, nome)
- SEXO (id, nome)
- REGIAO (id, nome)
- UF (id, id_regiao, sigla_uf)
  - id_regiao **REFERENCIA** REGIAO(id)
- EMPREGADO (id, id_ocupacao, id_sexo, id_uf, ano, remuneracao_media, desligamento, idade)
  - id_ocupacao **REFERENCIA** OCUPACAO(id)
  - id_sexo **REFERENCIA** SEXO(id)
  - id_uf **REFERENCIA** UF(id)

2.3 Modelo Físico

```
OCUPACAO (
    id INT NOT NULL,
    nome VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
);

SEXO (
    id INT NOT NULL,
    nome VARCHAR(9) NOT NULL,
    PRIMARY KEY (id)
);

REGIAO (
    id INT NOT NULL,
    nome VARCHAR(12) NOT NULL,
    PRIMARY KEY (id)
);

UF (
    id INT NOT NULL,
    id_regiao INT NOT NULL,
    nome VARCHAR(2) NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_regiao) REFERENCES REGIAO(id)
);

EMPREGADO (
    id INT NOT NULL AUTO_INCREMENT,
    id_ocupacao INT NOT NULL,
    id_sexo INT NOT NULL,
```

```
    id_uf INT NOT NULL,
    id_ano INT NOT NULL,
    remuneracao_media FLOAT,
    desligamento INT,
    idade INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_ocupacao) REFERENCES OCUPACAO(id),
    FOREIGN KEY (id_sexo) REFERENCES SEXO(id),
    FOREIGN KEY (id_ano) REFERENCES ANO(id),
    FOREIGN KEY (id_uf) REFERENCES UF(id)
);
```

3. O script SQL que gerou o banco de dados.

```
-- ============= CRIANDO DB ========================

DROP DATABASE IF EXISTS projfbd;
CREATE DATABASE projfbd DEFAULT CHARACTER SET 'utf8';
USE projfbd;

-- ============= CRIANDO AS TABELAS ========================

CREATE TABLE
    OCUPACAO (
        id INT NOT NULL,
        nome VARCHAR(255) NOT NULL,
        PRIMARY KEY (id)
    ) ENGINE = InnoDB;

CREATE TABLE
    SEXO (
        id INT NOT NULL,
        nome VARCHAR(9) NOT NULL,
        PRIMARY KEY (id)
    ) ENGINE = InnoDB;

CREATE TABLE
    REGIAO (
        id INT NOT NULL,
        nome VARCHAR(12) NOT NULL,
        PRIMARY KEY (id)
    ) ENGINE = InnoDB;

CREATE TABLE
    UF (
        id INT NOT NULL,
        id_regiao INT NOT NULL,
        nome VARCHAR(2) NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (id_regiao) REFERENCES REGIAO(id)
    ) ENGINE = InnoDB;

CREATE TABLE
    EMPREGADO (
        id INT NOT NULL AUTO_INCREMENT,
        id_ocupacao INT NOT NULL,
        id_sexo INT NOT NULL,
```

```
    id_uf INT NOT NULL,
    ano INT NOT NULL,
    remuneracao_media FLOAT,
    desligamento INT,
    idade INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (id_ocupacao) REFERENCES OCUPACAO(id),
    FOREIGN KEY (id_sexo) REFERENCES SEXO(id),
    FOREIGN KEY (id_uf) REFERENCES UF(id)
) ENGINE = InnoDB;
```

4. Apresentar o processo de ETL (Extract, Transform, Load) para importação dos dados para o banco de dados.

Os passos do ETL estão contidos nos arquivos abaixo:

- Juntando os anos de 2018 a 2021 em um único arquivo.
  - Arquivo: 03_juntando_todos_anos.ipynb
- Juntando os dados da RAIS com os dados de sexo e raça.
  - Arquivo: 04_fazendo_join_ocup_sexo.ipynb

Resumindo os passos do ETL

Extração

1. Após filtrar os dados da tabela Relação Anual de Informações Sociais (RAIS) pelos anos de 2018 a 2021, foi feito o filtro pelos IDs de cargos de Tecnologia da informação conforme a Classificação Brasileira de Ocupações (CBO).
   - 212205: Engenheiro de Aplicativos em Computacao
   - 212210: Engenheiro de Equipamentos em Computacao
   - 212215: Engenheiros de Sistemas Operacionais em Computacao
   - 212305: Administrador de Banco de Dados
   - 212310: Administrador de Redes
   - 212315: Administrador de Sistemas Operacionais
   - 212320: Administrador em Segurança da Informação
   - 212405: Analista de Desenvolvimento de Sistemas
   - 212410: Analista de Redes e de Comunicacao de Dados
   - 212415: Analista de Sistemas de Automacao
   - 212420: Analista de Suporte Computacional

    - 317105: Programador de Internet
    - 317110: Programador de Sistemas de Informacao
    - 317115: Programador de Maquinas - Ferramenta com Comando Numerico
    - 317120: Programador de Multimidia
    - 317205: Operador de Computador (Inclusive Microcomputador)
    - 317210: Tecnico de Apoio ao Usuario de Informatica (Helpdesk)
2. Em seguida, filtramos os dados de profissionais de TI entre os anos de 2018 a 2021

- Quantidade **total**: 1.543.009
- Quantidade **por ano**:
    - 2021: 691.982
    - 2018: 466.852
    - 2020: 222.102
    - 2019: 162.073

Transformação

1. Primeiro, juntamos os dados com a planilha de Sexo

```
1,Masculino
2,Feminino
-1,Ignorado
```

2. Em seguida, juntamos os dados com a planilha com o nome dos Cargos

```
212205,Engenheiro de Aplicativos em Computacao
212210,Engenheiro de Equipamentos em Computacao
...
```

3. Logo após, selecionamos apenas a colunas necessárias e as renomeamos

Carregamento

Fazendo INSERT dos dados das tabelas (todas menos a tabela EMPREGADO) no Banco:

```
-- Active: 1684245138463@@127.0.0.1@3306@projfbd

USE projfbd;

-- ============= INSERINDO OS DADOS (CARGA) ========================

INSERT INTO OCUPACAO (id, nome)
VALUES (
        212405,
        'Analista de Desenvolvimento de Sistemas'
    ), (
        317110,
        'Programador de Sistemas de Informacao'
    ), (
        212420,
```

```
        'Analista de Suporte Computacional'
    ), (
        317210,
        'Tecnico de Apoio ao Usuario de Informatica (Helpdesk)'
    ), (
        212410,
        'Analista de Redes e de Comunicacao de Dados'
    ), (
        317205,
        'Operador de Computador (Inclusive Microcomputador)'
    ), (
        212315,
        'Administrador de Sistemas Operacionais'
    ), (
        212415,
        'Analista de Sistemas de Automacao'
    ), (
        212310,
        'Administrador de Redes'
    ), (
        212205,
        'Engenheiro de Aplicativos em Computacao'
    ), (
        212305,
        'Administrador de Banco de Dados'
    ), (
        212320,
        'Administrador em Segurança da Informação'
    ), (
        317105,
        'Programador de Internet'
    ), (
        212215,
        'Engenheiros de Sistemas Operacionais em Computacao'
    ), (
        317115,
        'Programador de Maquinas - Ferramenta com Comando Numerico'
    ), (
        317120,
        'Programador de Multimidia'
    ), (
        212210,
        'Engenheiro de Equipamentos em Computacao'
    );

-- ==== TABELA: SEXO

INSERT INTO SEXO (id, nome) VALUES (1, 'Masculino'), (2, 'Feminino');

-- ==== TABELA: REGIAO

INSERT INTO REGIAO (id, nome)
VALUES (3, 'Sudeste'), (4, 'Sul'), (1, 'Nordeste'), (2, 'Norte'), (0, 'Centro-
Oeste');

-- ==== TABELA: UF

INSERT INTO
    UF (id, id_regiao, nome)
```

```
VALUES (25, 3, 'SP'), (18, 3, 'RJ'), (10, 3, 'MG'), (22, 4, 'RS'), (17, 4, 'PR'),
(23, 4, 'SC'), (6, 0, 'DF'), (20, 2, 'RO'), (15, 1, 'PE'), (4, 1, 'BA'), (5, 1,
'CE'), (7, 3, 'ES'), (8, 0, 'GO'), (12, 0, 'MT'), (13, 2, 'PA'), (14, 1, 'PB'), (2,
2, 'AM'), (11, 0, 'MS'), (19, 1, 'RN'), (9, 1, 'MA'), (16, 1, 'PI'), (1, 1, 'AL'),
(24, 1, 'SE'), (26, 2, 'TO'), (0, 2, 'AC'), (3, 2, 'AP'), (21, 2, 'RR');
```

A tabela de EMPREGADO foi feita com o script python abaixo, pois continha mais de 1.5 milhões de registros

In [ ]:

```python
from __future__ import print_function
import pandas as pd
import mysql.connector
from datetime import date, datetime, timedelta


class CargaFullTabelaRAIS:
  def __init__(self, batch_size=1000, size_max=10000):
    self.batch_size = batch_size
    self.size_max = size_max
    self.cnx = mysql.connector.connect(
        user='root', password='root', database='projfbd')
    self.cursor = self.cnx.cursor()
    self.path_file_parquet = "../output/gold/rais_TODOS_ANOS_comJoin_RAIS_VINC_PUB.parquet.gz

  def carregar_CSV(self):
      # Mostrar mais colunas
    pd.set_option("display.max_columns", 100)
    pd.set_option('display.max_colwidth', 100)

    df = pd.read_parquet(self.path_file_parquet)

    qnt_total = len(df)

    print(f"""
      Quantidade de docs carregados: {qnt_total}
      """)
    return df

  def get_rais_from_pandas(self):
      df = self.carregar_CSV()
      counts = df[[
          "ocupacao_id", "sexo_id", "sigla_uf_id", "ano", "remuneracao_media", "desligamento"
      lista = counts.values.tolist()
      lista = [tuple(x) for x in lista]

      return lista

  def gerar_batch_insert_rais(self):
    print(f"""
      Inserindo os dados...
      """)

    rais_from_pandas = self.get_rais_from_pandas()

    for i in range(0, len(rais_from_pandas[:self.size_max]), self.batch_size):
      add_query = ("INSERT INTO EMPREGADO "
                   "(id_ocupacao, id_sexo, id_uf, ano, remuneracao_media, desligamento, idade
                   "VALUES ")

      vals = ", ".join((f"({str(id_ocupacao)}, {str(id_sexo)}, {str(id_uf)}, {str(ano)}, {str
                        for id_ocupacao, id_sexo, id_uf, ano, remuneracao_media, desligamento,
      print(f"\n\n -- ====  INSERT a partir do dado {i}")
```

```
      # print(add_query + vals)
      try:
        self.cursor.execute(add_query + vals)
      except Exception as e:
        print(e)
        print(i)

    # ============ COMMIT ============
    self.cnx.commit()

  def init(self):
    self.gerar_batch_insert_rais()

  def __del__(self):
    self.cursor.close()
    self.cnx.close()


# ================ INICIANDO

CargaFullTabelaRAIS(batch_size=500000, size_max=2000000).init()
```

```
    Inserindo os dados...


    Quantidade de docs carregados: 1543009



-- ====  INSERT a partir do dado 0


-- ====  INSERT a partir do dado 500000


-- ====  INSERT a partir do dado 1000000


-- ====  INSERT a partir do dado 1500000
```

5. Utilização de pelo menos uma View.

```
CREATE VIEW VW_EMPREGADO_FULL AS
    SELECT
        e.*,
        OCUPACAO.nome AS ocupacao,
        SEXO.nome AS sexo,
        UF.nome AS uf,
        REGIAO.nome AS regiao
    FROM EMPREGADO as e
        INNER JOIN OCUPACAO ON e.id_ocupacao = OCUPACAO.id
        INNER JOIN SEXO ON e.id_sexo = SEXO.id
        INNER JOIN UF ON e.id_uf = UF.id
        INNER JOIN REGIAO ON UF.id_regiao = REGIAO.id;
```

In [ ]:
```
df_5 = pd.read_sql("""
SELECT * FROM VW_EMPREGADO_FULL
LIMIT 5;
""", cnx)
```

```
df_5
```

Out[ ]:

	id	id_ocupacao	id_sexo	id_uf	ano	remuneracao_media	desligamento	idade	ocupacao
0	7293	212410	1	6	2019	30772.5	0	61	Analista de Redes e de Comunicacao de Dados
1	7292	212410	1	6	2019	24142.7	0	56	Analista de Redes e de Comunicacao de Dados
2	7291	212410	1	6	2019	24667.7	0	58	Analista de Redes e de Comunicacao de Dados
3	7290	212410	1	6	2019	23558.3	0	54	Analista de Redes e de Comunicacao de Dados
4	7289	212410	1	6	2019	24537.7	0	57	Analista de Redes e de Comunicacao de Dados

6. Utilização de pelo menos uma Procedure (com comandos condicionais)

```
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS PROC_SELECIONAR_EMP_POR_ANO (IN ANO INT)
BEGIN
    SELECT * FROM `VW_EMPREGADO_FULL` as e WHERE ANO = e.ano;
END;

$$
DELIMITER;
```

In [ ]:

```
df_6 = pd.read_sql("""
CALL PROC_SELECIONAR_EMP_POR_ANO(2018);
""", cnx).head(5)

df_6
```

Out[ ]:

	id	id_ocupacao	id_sexo	id_uf	ano	remuneracao_media	desligamento	idade	ocupac
0	391322	212405	1	12	2018	10061.30	1	35	Analista Desenvolvime de Sister
1	391321	212315	1	12	2018	14284.80	1	34	Administra de Sister Operacior
2	391320	212320	1	12	2018	8000.01	1	45	Administra em Segurança Informa
3	391319	212215	1	12	2018	8000.01	1	39	Engenheiros Sister Operacionais Computa
4	391318	212405	1	12	2018	4003.71	1	39	Analista Desenvolvime de Sister

7. Utilização de pelo menos um trigger (com comandos condicionais)

```
DELIMITER $$

CREATE TRIGGER IF NOT EXISTS TRIGGER_CHECK_UPDATE_EMPREGADO
BEFORE UPDATE ON EMPREGADO FOR EACH ROW BEGIN
    IF NEW.idade < 14 THEN SET NEW.idade = OLD.idade;
    END IF;
END;

$$

DELIMITER;
```

Tentando alterar idade para menos que 14 anos (conforme restrição da TRIGGER)

In [ ]:

```
cnx = get_cnx()
cursor = cnx.cursor()
cursor.execute(("UPDATE EMPREGADO "
                "SET idade = 11 "
                "WHERE id = 10"
                ))

df_7_1 = pd.read_sql("""
SELECT * FROM EMPREGADO WHERE id = 10
""", cnx).tail(1)

df_7_1
```

Out[ ]:

	id	id_ocupacao	id_sexo	id_uf	ano	remuneracao_media	desligamento	idade
0	10	317210	1	11	2019	1640.49	0	31

Tentando alterar idade para mais que 14 anos. Desta vez funcionou.

In [ ]:
```
cnx = get_cnx()
cursor = cnx.cursor()
cursor.execute(("UPDATE EMPREGADO "
                "SET idade = 40 "
                "WHERE id = 10"
                ))

df_7_2 = pd.read_sql("""
SELECT * FROM EMPREGADO WHERE id = 10
""", cnx).tail(1)

df_7_2
```

Out[ ]:

	id	id_ocupacao	id_sexo	id_uf	ano	remuneracao_media	desligamento	idade
0	10	317210	1	11	2019	1640.49	0	40

8. No mínimo 5 Consultas SQL (a complexidade da consulta será avaliada)

8.1 Quantidade de empregos de TI por ano e gênero

```
SELECT ano, sexo, COUNT(*) as qnt_empregos
FROM `VW_EMPREGADO_FULL`
GROUP BY ano, sexo;
```

In [ ]:
```
df_8_1 = pd.read_sql("""
SELECT ano, sexo, COUNT(*) as qnt_empregos
FROM `VW_EMPREGADO_FULL`
GROUP BY ano, sexo;
""", cnx)

df_8_1
```

8.2 Quantidade de cargos por gênero

```
SELECT
    ocupacao,
    sexo,
    COUNT(id_ocupacao) as qnt_cargos
FROM `VW_EMPREGADO_FULL`
GROUP BY id_ocupacao, sexo
ORDER BY ocupacao, sexo;
```

In [ ]:
```
df_8_2 = pd.read_sql("""
    SELECT
        ocupacao,
        sexo,
        COUNT(id_ocupacao) as qnt_cargos
    FROM `VW_EMPREGADO_FULL`
    GROUP BY id_ocupacao, sexo
    ORDER BY ocupacao, sexo;
""", cnx)
```

```
df_8_2
```

Out[ ]:

	ocupacao	sexo	qnt_cargos
0	Administrador de Banco de Dados	Feminino	6116
1	Administrador de Banco de Dados	Masculino	24066
2	Administrador de Redes	Feminino	3970
3	Administrador de Redes	Masculino	33534
4	Administrador de Sistemas Operacionais	Feminino	10380
5	Administrador de Sistemas Operacionais	Masculino	37810
6	Administrador em Segurança da Informação	Feminino	5858
7	Administrador em Segurança da Informação	Masculino	23500
8	Analista de Desenvolvimento de Sistemas	Feminino	216176
9	Analista de Desenvolvimento de Sistemas	Masculino	890878
10	Analista de Redes e de Comunicacao de Dados	Feminino	51756
11	Analista de Redes e de Comunicacao de Dados	Masculino	149734
12	Analista de Sistemas de Automacao	Feminino	8580
13	Analista de Sistemas de Automacao	Masculino	38404
14	Analista de Suporte Computacional	Feminino	96538
15	Analista de Suporte Computacional	Masculino	383148
16	Engenheiro de Aplicativos em Computacao	Feminino	4144
17	Engenheiro de Aplicativos em Computacao	Masculino	26470
18	Engenheiro de Equipamentos em Computacao	Feminino	666
19	Engenheiro de Equipamentos em Computacao	Masculino	4882
20	Engenheiros de Sistemas Operacionais em Computacao	Feminino	2680
21	Engenheiros de Sistemas Operacionais em Computacao	Masculino	19490
22	Operador de Computador (Inclusive Microcomputador)	Feminino	47552
23	Operador de Computador (Inclusive Microcomputador)	Masculino	118982
24	Programador de Internet	Feminino	4156
25	Programador de Internet	Masculino	23268
26	Programador de Maquinas - Ferramenta com Comando Numerico	Feminino	1078
27	Programador de Maquinas - Ferramenta com Comando Numerico	Masculino	17510
28	Programador de Multimidia	Feminino	2050
29	Programador de Multimidia	Masculino	9184
30	Programador de Sistemas de Informacao	Feminino	112974
31	Programador de Sistemas de Informacao	Masculino	416248
32	Tecnico de Apoio ao Usuario de Informatica (Helpdesk)	Feminino	64100
33	Tecnico de Apoio ao Usuario de Informatica (Helpdesk)	Masculino	230136

8.3 Remuneração média por região

```
WITH remun_media AS (
      SELECT
          regiao,
          sexo,
          ROUND(MIN(remuneracao_media), 2) as minimo,
          ROUND(MAX(remuneracao_media), 2) as maximo,
          ROUND(STD(remuneracao_media), 2) as desvio_padrao,
          ROUND(AVG(remuneracao_media), 2) as media
      FROM
          `VW_EMPREGADO_FULL`
      WHERE
          remuneracao_media > 0
      GROUP BY
          regiao,
          sexo
  )
SELECT
  *,
  ROUND(
      media - LAG(media, 1) OVER (
          PARTITION BY regiao
          ORDER BY
              regiao,
              sexo
      ),
      2
  ) as diff_media_por_regiao
FROM remun_media
ORDER BY regiao, sexo;
```

In [ ]:
```
df_8_3 = pd.read_sql(f"""
WITH remun_media AS (
      SELECT
          regiao,
          sexo,
          ROUND(MIN(remuneracao_media), 2) as minimo,
          ROUND(MAX(remuneracao_media), 2) as maximo,
          ROUND(STD(remuneracao_media), 2) as desvio_padrao,
          ROUND(AVG(remuneracao_media), 2) as media
      FROM
          `VW_EMPREGADO_FULL`
      WHERE
          remuneracao_media > 0
      GROUP BY
          regiao,
          sexo
  )
SELECT
  *,
  ROUND(
      media - LAG(media, 1) OVER (
          PARTITION BY regiao
          ORDER BY
              regiao,
              sexo
      ),
      2
```

```
  ) as diff_media_por_regiao
FROM remun_media
ORDER BY regiao, sexo;
""", cnx)

df_8_3
```

Out[ ]:

	regiao	sexo	minimo	maximo	desvio_padrao	media	diff_media_por_regiao
0	Centro-Oeste	Feminino	315.69	73685.48	7033.05	6542.79	NaN
1	Centro-Oeste	Masculino	300.00	85750.97	6985.15	7186.17	643.38
2	Nordeste	Feminino	313.86	44387.25	3957.06	3900.37	NaN
3	Nordeste	Masculino	291.29	98543.80	3694.91	3902.51	2.14
4	Norte	Feminino	287.07	110716.20	2904.23	3060.56	NaN
5	Norte	Masculino	287.53	105844.80	4050.40	4062.80	1002.24
6	Sudeste	Feminino	289.87	146400.00	4765.07	5293.50	NaN
7	Sudeste	Masculino	288.80	161407.66	5462.57	6094.63	801.13
8	Sul	Feminino	302.60	88432.50	3672.33	4123.61	NaN
9	Sul	Masculino	288.14	84126.00	3856.57	4745.27	621.66

8.4 Quantidade de demissões por região

```sql
WITH qnt_desligs AS (
        SELECT
            ano,
            regiao,
            sexo,
            COUNT(desligamento) as qnt_desligamento
        FROM
            `VW_EMPREGADO_FULL`
        GROUP BY
            ano,
            regiao,
            sexo
    )
SELECT
    *,
    ROUND(
        qnt_desligamento - LAG(qnt_desligamento, 1) OVER (
            PARTITION BY ano, regiao
            ORDER BY
                ano,
                regiao,
                sexo
        ),
        2
    ) as diff_deslig_por_ano_regiao
FROM qnt_desligs as q
ORDER BY ano, regiao, sexo;
```

In [ ]:

```
df_8_4 = pd.read_sql(f"""
WITH qnt_desligs AS (
```

```
        SELECT
            ano,
            regiao,
            sexo,
            COUNT(desligamento) as qnt_desligamento
        FROM
            `VW_EMPREGADO_FULL`
        GROUP BY
            ano,
            regiao,
            sexo
    )
SELECT
    *,
    ROUND(
        qnt_desligamento - LAG(qnt_desligamento, 1) OVER (
            PARTITION BY ano, regiao
            ORDER BY
                ano,
                regiao,
                sexo
        ),
        2
    ) as diff_deslig_por_ano_regiao
FROM qnt_desligs as q
ORDER BY ano, regiao, sexo;
""", cnx)

df_8_4
```

Out[ ]:

	ano	regiao	sexo	qnt_desligamento	diff_deslig_por_ano_regiao
0	2018	Centro-Oeste	Feminino	23073	NaN
1	2018	Centro-Oeste	Masculino	83871	60798.0
2	2018	Nordeste	Feminino	23880	NaN
3	2018	Nordeste	Masculino	107175	83295.0
4	2018	Norte	Feminino	82827	NaN
5	2018	Norte	Masculino	94455	11628.0
6	2018	Sudeste	Feminino	137667	NaN
7	2018	Sudeste	Masculino	630225	492558.0
8	2018	Sul	Feminino	37527	NaN
9	2018	Sul	Masculino	179856	142329.0
10	2019	Centro-Oeste	Feminino	9957	NaN
11	2019	Centro-Oeste	Masculino	52083	42126.0
12	2019	Nordeste	Feminino	5793	NaN
13	2019	Nordeste	Masculino	31893	26100.0
14	2019	Norte	Feminino	2496	NaN
15	2019	Norte	Masculino	5400	2904.0
16	2019	Sudeste	Feminino	47835	NaN
17	2019	Sudeste	Masculino	234573	186738.0
18	2019	Sul	Feminino	11583	NaN
19	2019	Sul	Masculino	84606	73023.0
20	2020	Centro-Oeste	Feminino	9486	NaN
21	2020	Centro-Oeste	Masculino	49371	39885.0
22	2020	Nordeste	Feminino	12237	NaN
23	2020	Nordeste	Masculino	54408	42171.0
24	2020	Norte	Feminino	12327	NaN
25	2020	Norte	Masculino	12756	429.0
26	2020	Sudeste	Feminino	75864	NaN
27	2020	Sudeste	Masculino	321165	245301.0
28	2020	Sul	Feminino	24057	NaN
29	2020	Sul	Masculino	94635	70578.0
30	2021	Centro-Oeste	Feminino	25437	NaN
31	2021	Centro-Oeste	Masculino	120456	95019.0
32	2021	Nordeste	Feminino	35613	NaN
33	2021	Nordeste	Masculino	139062	103449.0
34	2021	Norte	Feminino	6852	NaN

	ano	regiao	sexo	qnt_desligamento	diff_deslig_por_ano_regiao
35	2021	Norte	Masculino	37773	30921.0
36	2021	Sudeste	Feminino	296853	NaN
37	2021	Sudeste	Masculino	1045752	748899.0
38	2021	Sul	Feminino	76797	NaN
39	2021	Sul	Masculino	291351	214554.0

8.5 Quantidades de demissões com dados acumulados por ano

```sql
WITH qnt_desligs AS (
        SELECT
            ano,
            sexo,
            COUNT(desligamento) as qnt_desligamento
        FROM
            `VW_EMPREGADO_FULL`
        GROUP BY ano, sexo
    )
SELECT
    *,
    ROUND(
        qnt_desligamento + LAG(qnt_desligamento, 1) OVER (
            PARTITION BY ano
            ORDER BY
                ano,
                sexo
        ),
        2
    ) as cum
FROM qnt_desligs as q
ORDER BY ano, sexo;
```

In [ ]:
```sql
df_8_5 = pd.read_sql(f"""
WITH qnt_desligs AS (
        SELECT
            ano,
            sexo,
            COUNT(desligamento) as qnt_desligamento
        FROM
            `VW_EMPREGADO_FULL`
        GROUP BY ano, sexo
    )
SELECT
    *,
    ROUND(
        qnt_desligamento + LAG(qnt_desligamento, 1) OVER (
            PARTITION BY ano
            ORDER BY
                ano,
                sexo
        ),
        2
    ) as cum
FROM qnt_desligs as q
```

```
ORDER BY ano, sexo;
""", cnx)

df_8_5
```

Out[ ]:

	ano	sexo	qnt_desligamento	cum
0	2018	Feminino	304974	NaN
1	2018	Masculino	1095582	1400556.0
2	2019	Feminino	77664	NaN
3	2019	Masculino	408555	486219.0
4	2020	Feminino	133971	NaN
5	2020	Masculino	532335	666306.0
6	2021	Feminino	441552	NaN
7	2021	Masculino	1634394	2075946.0

In [ ]:

```
# Montar Apresentação

# Convertendo tabelas para Markdown

# def to_mk(df):
#   print(df.to_markdown(index=False))
#   print("\n")

# to_mk(df_5)
# to_mk(df_6)
# to_mk(df_7_1)
# to_mk(df_7_2)
# to_mk(df_8_1)
# to_mk(df_8_2)
# to_mk(df_8_3)
# to_mk(df_8_4)
# to_mk(df_8_5)

# Exibindo url das imagens
# print(img_mm_1, "\n", img_mm_2)
```