

# Script\_Criacao\_Banco\_Dados

October 10, 2021

**Trabalho:** Projeto Integrador 3

**Discentes:** Bruna Mattioli de Oliveira e Gabriel Andrade Varga

**Descrição:** Criação do Banco de Dados do Data Warehouse no PostgreSQL e imputação dos dados oriundos de um arquivo CSV diretamente no Data Warehouse

```
[1]: import numpy as np
import pandas as pd
import psycopg2
import csv
import pandas.io.sql as sqlio
from matplotlib import pyplot as plt
from matplotlib import ticker as ticker
import locale
locale.setlocale(locale.LC_NUMERIC, "pt_BR.UTF-8")
```

```
[1]: 'pt_BR.UTF-8'
```

Importação do Banco de Dados em CSV e Criação das Tabelas no DW

```
[2]: # Conexão no postgre com o banco de dados Projeto_Integrador_3
conn = psycopg2.connect("dbname=Projeto_Integrador_3 user=postgres_
    ↳password=vagan9ch")
cur = conn.cursor()

# Criação das tabelas modeladas no Postgre
cur.execute("CREATE TABLE vendedor_loja (id_vendedor_loja SERIAL PRIMARY KEY,
    ↳nm_vendedor VARCHAR, nm_loja VARCHAR, cidade_loja VARCHAR, estado_loja_
    ↳VARCHAR, regioao_loja VARCHAR);")
cur.execute("CREATE TABLE cliente (id_cliente SERIAL PRIMARY KEY,
    ↳cidade_cliente VARCHAR, estado_cliente VARCHAR, uf_cliente VARCHAR,
    ↳faixa_idade VARCHAR, ocupacao VARCHAR);")
cur.execute("CREATE TABLE inadimplencia (id_inadimplencia SERIAL PRIMARY KEY,
    ↳fg_pg_em_dia INTEGER, fg_atraso_30dias INTEGER, fg_atraso_60dias INTEGER,
    ↳fg_atraso_90dias INTEGER);")
```

```

cur.execute("CREATE TABLE dados_veiculo (id_veiculo SERIAL PRIMARY KEY,
↳cilindradas NUMERIC(30,2), qtd_portas INTEGER, qtd_assentos INTEGER,
↳marca_veiculo VARCHAR, modelo_veiculo VARCHAR, tipo_veiculo VARCHAR,
↳valor_veiculo NUMERIC(30,2), vetor_caracteristica FLOAT[]);")
cur.execute("CREATE TABLE tempo (id_tempo SERIAL PRIMARY KEY, mes VARCHAR, ano
↳INTEGER);")
cur.execute("CREATE TABLE dados_financiamento (id_dados_financiamento SERIAL
↳PRIMARY KEY, prazo VARCHAR, faixa_score VARCHAR);")
cur.execute("CREATE TABLE contratos_fato (id_contrato SERIAL PRIMARY KEY,
↳id_inadimplencia INTEGER, id_vendedor_loja INTEGER, id_dados_financiamento
↳INTEGER, id_tempo INTEGER, id_cliente INTEGER, id_veiculo INTEGER,
↳vlr_financiado NUMERIC(30,2), vlr_entrada NUMERIC(30,2), vlr_juros
↳NUMERIC(30,2), vlr_operacao NUMERIC(30,2), qtd_veiculos INTEGER);")

# Atribuição das chaves estrangeiras
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY (id_inadimplencia)
↳REFERENCES inadimplencia (id_inadimplencia);")
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY (id_vendedor_loja)
↳REFERENCES vendedor_loja (id_vendedor_loja);")
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY
↳(id_dados_financiamento) REFERENCES dados_financiamento
↳(id_dados_financiamento);")
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY (id_tempo) REFERENCES
↳tempo (id_tempo);")
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY (id_cliente) REFERENCES
↳cliente (id_cliente);")
cur.execute("ALTER TABLE contratos_fato ADD FOREIGN KEY (id_veiculo) REFERENCES
↳dados_veiculo (id_veiculo);")

# Execução das atualizações
conn.commit()

```

## Imputação dos Dados no PostgreSQL

```

[3]: #-----
# Imputando os dados da tabela / entidade vendedor_loja no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↳Bases/tabela_vendedor_loja.csv',newline='', encoding='utf8',
↳errors='ignore') as csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['estado_loja'] != '':
            insert = "INSERT INTO vendedor_loja(nm_vendedor, nm_loja,
↳cidade_loja, estado_loja, regioao_loja) VALUES (%s,%s,%s,%s,%s)"

```

```

        parametros = (row['nm_vendedor'], row['nm_loja'],
        ↪row['cidade_loja'], row['estado_loja'], row['regiao_loja'])
        cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()

```

```

[4]: #-----
# Imputando os dados da tabela / entidade cliente no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↪Bases/tabela_cliente.csv',newline='', encoding='utf8', errors='ignore') as
↪csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['uf_cliente'] != '':
            insert = "INSERT INTO cliente(cidade_cliente, estado_cliente,
            ↪uf_cliente, ocupacao, faixa_idade) VALUES (%s,%s,%s,%s,%s)"
            parametros = (row['cidade_cliente'], row['estado_cliente'],
            ↪row['uf_cliente'], row['ocupacao'], row['faixa_idade'])
            cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()

```

```

[5]: #-----
# Imputando os dados da tabela / entidade inadimplencia no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↪Bases/tabela_inadimplencia.csv',newline='', encoding='utf8',
↪errors='ignore') as csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['fg_pg_em_dia'] != '':
            insert = "INSERT INTO inadimplencia(fg_pg_em_dia, fg_atraso_30dias,
            ↪fg_atraso_60dias, fg_atraso_90dias) VALUES (%s,%s,%s,%s)"
            parametros = (row['fg_pg_em_dia'], row['fg_atraso_30dias'],
            ↪row['fg_atraso_60dias'], row['fg_atraso_90dias'])
            cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()

```

```
[6]: #-----
# Imputando os dados da tabela / entidade dados_veiculo no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↳Bases/tabela_dados_veiculos.csv',newline='', encoding='utf8',
↳errors='ignore') as csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['modelo_veiculo'] != '':
            insert = "INSERT INTO dados_veiculo(cilindradas, qtd_portas,
↳qtd_assentos, marca_veiculo, modelo_veiculo, tipo_veiculo, valor_veiculo,
↳vetor_caracteristica) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
            parametros = (row['cilindradas'], row['qtd_portas'],
↳row['qtd_assentos'], row['marca_veiculo'], row['modelo_veiculo'],
↳row['tipo_veiculo'], row['valor_veiculo'], row['vetor_caracteristica'])
            cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()
```

```
[7]: #-----
# Imputando os dados da tabela / entidade tempo no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↳Bases/tabela_tempo.csv',newline='', encoding='utf8', errors='ignore') as
↳csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['mes'] != '':
            insert = "INSERT INTO tempo(mes, ano) VALUES (%s,%s)"
            parametros = (row['mes'], row['ano'])
            cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()
```

```
[8]: #-----
# Imputando os dados da tabela / entidade dados_financiamento no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↳Bases/tabela_dados_financiamento.csv',newline='', encoding='utf8',
↳errors='ignore') as csvfile:
```

```

reader = csv.DictReader(csvfile, delimiter=';')
for row in reader:
    if row['prazo'] != '':
        insert = "INSERT INTO dados_financiamento(prazo, faixa_score)
↪VALUES (%s,%s)"
        parametros = (row['prazo'], row['faixa_score'])
        cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()

```

```

[11]: #-----
# Imputando os dados da tabela / entidade contratos_fato no DW
#-----

# Imputando os dados via um looping no CSV linha por linha
with open('G:/My Drive/Especialização/Disciplinas/Módulo 3/Projeto Integrador 3/
↪Bases/tabela_fato.csv',newline='', encoding='utf8', errors='ignore') as
↪csvfile:
    reader = csv.DictReader(csvfile, delimiter=';')
    for row in reader:
        if row['vlr_financiado'] != '':
            id_vendedor_loja = 0
            id_cliente = 0
            id_inadimplencia = 0
            id_veiculo = 0
            id_tempo = 0
            id_dados_financiamento = 0

            query1 = "SELECT id_vendedor_loja FROM vendedor_loja WHERE
↪id_vendedor_loja = '%s'"
            var1 = row['id_vendedor_loja']

            query2 = "SELECT id_cliente FROM cliente WHERE id_cliente = '%s'"
            var2 = row['id_cliente']

            query3 = "SELECT id_inadimplencia FROM inadimplencia WHERE
↪id_inadimplencia = '%s'"
            var3 = row['id_inadimplencia']

            query4 = "SELECT id_veiculo FROM dados_veiculo WHERE id_veiculo =
↪'%s'"
            var4 = row['id_veiculo']

            query5 = "SELECT id_tempo FROM tempo WHERE id_tempo = '%s'"
            var5 = row['id_tempo']

```

```

        query6 = "SELECT id_dados_financiamento FROM dados_financiamento_
↪WHERE id_dados_financiamento = '%s'"
        var6 = row['id_dados_financiamento']

        cur.execute(query1 % var1)
        id_vendedor_loja = cur.fetchone()

        cur.execute(query2 % var2)
        id_cliente = cur.fetchone()

        cur.execute(query3 % var3)
        id_inadimplencia = cur.fetchone()

        cur.execute(query4 % var4)
        id_veiculo = cur.fetchone()

        cur.execute(query5 % var5)
        id_tempo = cur.fetchone()

        cur.execute(query6 % var6)
        id_dados_financiamento = cur.fetchone()

        insert = "INSERT INTO contratos_fato(id_inadimplencia,
↪id_vendedor_loja, id_dados_financiamento, id_tempo, id_cliente, id_veiculo,
↪vlr_financiado, vlr_entrada, vlr_juros, vlr_operacao, qtd_veiculos) VALUES
↪(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        parametros = (id_inadimplencia[0], id_vendedor_loja[0],
↪id_dados_financiamento[0], id_tempo[0], id_cliente[0], id_veiculo[0],
↪row['vlr_financiado'], row['vlr_entrada'], row['vlr_juros'],
↪row['vlr_operacao'], row['qtd_veiculos'])
        cur.execute(insert, parametros)

# Execução das atualizações
conn.commit()

```