desafio_solucao

June 4, 2023

[1]:
```
! pip install -q pandas
```

```
[notice] A new release of pip
available: 22.3.1 -> 23.1.2
[notice] To update, run:
pip install --upgrade pip
```

[2]:
```
import pandas as pd
```

[7]:
```
vendas = pd.read_csv("data/base_vendas.csv", sep=";")
vendas = vendas.drop(columns=["Unnamed: 10"])
vendas.head()
```

[7]:

	data_solicitacao	data_venda	data_pagto	valor_venda	pol_comissao	valor_comissao	estado_cliente	estado_vendedor	produto	vendedor_id
0	13/01/2021	12/01/2021	01/02/2021	un.	750.00	SC	PR	A	V_1	NaN
1	14/02/2021	16/02/2021	01/03/2021	un.	750.00	SC	PR	A	V_1	NaN
2	17/02/2021	18/02/2021	01/03/2021	un.	750.00	PR	A	V_23	NaN	NaN
3	21/03/2021	25/03/2021	01/04/2021	un.	750.00	PR	A	V_23	NaN	NaN
4	29/01/2021	31/01/2021	01/03/2021	un.	750.00	PR	A	V_30	NaN	NaN

[8]:
```
vendedores = pd.read_json("data/vendedores.json")
vendedores.head()
```

[8]:

	vendedor_id	data_de_entrada	idade	carga_horaria_diaria
0	V_1	14/06/2016	27	8
1	V_10	15/09/2011	50	6
2	V_11	23/07/2011	47	6
3	V_12	16/09/2016	43	6
4	V_13	11/08/2014	25	8

```
            escolaridade  salario_bruto
0  Ensino Médio Completo   R$ 1,750.00
1  Ensino Médio Completo   R$ 1,750.00
2  Ensino Médio Completo   R$ 1,750.00
3  Ensino Médio Completo   R$ 1,750.00
4  Ensino Médio Completo   R$ 1,750.00
```

[9]:
```
produtos = pd.read_json("data/produtos.json")
produtos.head()
```

[9]:

	produto_id	preco_base	desconto_maximo	custo_de_producao
0	A	5000	20%	1300
1	B	112000	50%	30000
2	C	2000	50%	600
3	D	1900	10%	500
4	E	40000	20%	10000

1 Exercício 1.1

Quais os problemas de negócio o dataset poderia resolver? Existem problemas relacionados à consistência dos dados no dataset? Como você resolveria tais problemas?

Pelo nome das tabelas, parece que a tabela "vendas" seria uma tabela fato registrando eventos de venda, com os datasets "produtos" e "vendedores" servindo como tabelas dimensão. No entanto, notei diversos problemas em como a tabela fato "vendas" foi construída, conforme indicado abaixo.

Como solução imediata, verificaria se o problema está na produção ou na extração do dado. É importante que esse processo de solução seja automatizado o máximo possível, com testes automáticos sobre a qualidade dos dados, envio de alertas, definição de data owners, etc.

1.0.1 Problemas de qualidade nos dados

- As colunas "produto" e "vendedor_id", que me parecem ser chaves estrangeiras estrangeiras para as tabelas dimensão, estão com problemas de integridade. Estão com referências inválidas para as suas tabelas dimensão ou até mesmo não possuem referências.

[8]:
```
print(set(vendas["produto"]).difference(set(produtos["produto_id"])),␣
↪end="\n\n")
print(set(vendas["vendedor_id"]).difference(set(vendedores["vendedor_id"])))
```

```
{'V_45', 'V_34', 'V_11', 'V_46', 'V_24', 'V_26', 'V_1', 'V_10', 'V_22', 'V_2',
'V_16', 'V_25', 'V_15', nan, 'V_6', 'V_3', 'V_40', 'V_19', 'V_9', 'V_18',
'V_41', 'V_12', 'V_36', 'V_4', 'V_13', 'V_8'}

{nan}
```

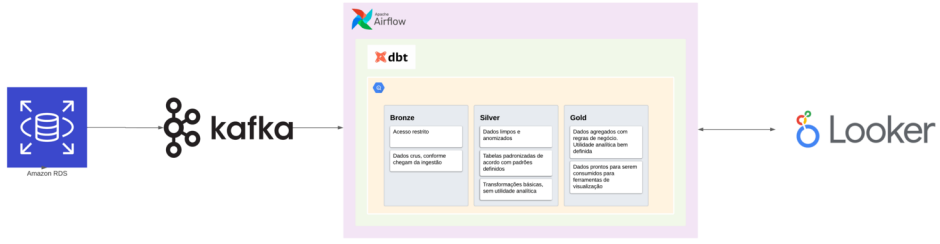
- As colunas "valor_venda", "pol_comissao", "valor_comissao", "estado_cliente", "estado_vendedor" estão com problemas de validade, possuindo valores fora do domínio de cada coluna. Parece que a matriz de valores foi embaralhada, misturando os valores.

[9]:
```
for column in ["valor_venda", "pol_comissao", "valor_comissao",␣
 ↪"estado_cliente", "estado_vendedor"]:
    print(f"{column}: {vendas[column].unique()}\n")
```

```
valor_venda: ['un.' '1.0%' '432000.00' '620000.00' '744000.00' '780000.00'
'910000.00'
 '936000.00' '1092000.00' '2000.00' '2400.00' '3200.00' '3840.00'
 '4500.00' '5400.00' '7250.00' '7400.00' '8700.00' '8750.00' '8880.00'
 '9600.00' '10500.00' '11520.00' '12000.00' '14400.00' '2.6%' '17400.00'
 '80000.00' '96000.00' '120000.00' '144000.00' '150000.00' '4.0%'
 '180000.00' '216000.00']

pol_comissao: ['750.00' '2800.00' '3360.00' '3600.00' '1.0%' '2.6%' nan '100.00'
 '450.00' '4.0%' '6000.00']

valor_comissao: ['SC' 'PR' 'RJ' 'SP' 'RS' 'MG' '4320.00' '6200.00' nan '7800.00'
'9360.00'
 '10920.00' '52.00' '62.40' 'BA' '99.84' '117.00' '140.40' '188.50'
 '226.20' '227.50' '249.60' '273.00' '312.00' '374.40' '452.40' 'AL' 'PE'
 'AC' '3200.00' '3840.00' '4800.00' '5760.00' '6000.00' '7200.00'
 '8640.00']

estado_cliente: ['PR' 'A' 'RS' 'SC' 'SP' 'MG' 'B' 'PE' 'BA' 'RJ' 'C' 'F' 'D'
'AC' 'AL']

estado_vendedor: ['A' 'V_23' 'V_30' 'V_20' 'V_22' 'V_25' 'V_24' 'B' 'V_47' 'SC'
'C' 'SP'
 'MG' 'V_32' 'V_17' 'V_31' 'V_34' 'V_21' 'F' 'V_39' 'V_27' 'D' 'V_26'
 'V_19' 'E' 'RS']
```

2 Exercício 1.2

Elabore uma arquitetura (desenho) adequada para a disponibilização desses dados para o consumo da área de Business Intelligence



3 Exercício 2

Considerando o diagrama de tabelas abaixo, monte as consultas em SQL que resultam em:

A lista de alunos e o montante de mensalidade de cada um deles, somente daqueles que não possuem nenhuma mensalidade com o status = 'EM ATRASO'

```
WITH
    tabelao as (
        SELECT
            ma.* AS matricula,
            me.* AS mensalidade,
            a.* AS aluno
        FROM matricula AS ma
            INNER JOIN mensalidade AS me ON ma.id_matricula = me.id
            INNER JOIN aluno AS a ON ma.id_aluno = a.id_aluno
    ),
    alunos_inadimplentes as (
        SELECT DISTINCT(t.aluno.id) AS id
        FROM tabelao AS t
        WHERE t.mensalidade = 'EM ATRASO'
    )
SELECT
    t.alunos.id,
    SUM(t.mensalidade.valor) AS montante_mensalidade
FROM tabelao AS t
LEFT JOIN alunos_inadimplentes AS ai ON t.aluno.id = ai.id
WHERE ai.id IS NULL
GROUP BY t.alunos.id
```

A lista com o nome dos alunos que possuem sua primeira matrícula na instituição no semestre atual, estão matriculados na disciplina 'Física I' e não possuem mensalidades registradas no sistema.

```
WITH
    alunos_novos as (
        SELECT
            a.id AS id
        FROM matricula AS ma
            INNER JOIN aluno AS a ON ma.id_aluno = a.id_aluno
        WHERE CURRENT_DATE() >= ma.periodo_letivo_inicio
                AND CURRENT_DATE() <= ma.periodo_letivo_fim
        GROUP BY a.id
        HAVING COUNT(ma.dt_matricula) = 1
    ),
    matriculas_em_fisica_1 AS (
        SELECT
            ma.id AS id
        FROM matricula AS ma
            INNER JOIN turma AS a ON ma.id_turma = t.id
```

```
            INNER JOIN disciplina AS d ON t.id_disciplina = d.id
        WHERE d.nome = 'Física I'
    )
SELECT
    a.nome
FROM matriculas_em_fisica_1 AS ma
    INNER JOIN alunos_novos AS an ON ma.id_aluno = an.id
    INNER JOIN aluno AS a ON an.id = a.id
    LEFT JOIN mensalidade AS me ON ma.id_matricula = me.id
WHERE me.id IS NULL
```

[ ]: