
FIAP Challenge Sofisa - Demonstração do Teradata Vantage



Cesar Augusto da Silva - educador.cesar@gmail.com

Lucas Alexandre Alves da Silva - lucas.alexandre1@hotmail.com

Marivaldo dos Santos Barbosa - marivaldo.barbosa@icloud.com

Thiago Alberto de Souza Colen - thiago.souzacolen@gmail.com

Como protótipo da utilização da arquitetura escolhida pela equipe, vamos demonstrar neste Jupyter Notebook um exemplo de utilização do Teradata Vantage e sua facilidade em gerar insights através de consultas à dados digitais, armazenados em Object Storage e também a partir de algumas de suas funcionalidades, como por exemplo a função NPATH

(<https://docs.teradata.com/r/aKnWloeEx3pmus0aivWOaw/MOSRYZ0og7DX6xFvY5X5KA>

(<https://docs.teradata.com/r/aKnWloeEx3pmus0aivWOaw/MOSRYZ0og7DX6xFvY5X5KA>)).

A função nPath verifica um conjunto de linhas, procurando os padrões que você especifica. Para cada conjunto de linhas de entrada que correspondem ao padrão, nPath produz uma única linha de saída. A função fornece um recurso flexível de correspondência de padrões que permite especificar padrões complexos nos dados de entrada e definir os valores de saída para cada conjunto de entrada correspondido.

Como funciona a função nPath?

nPath é útil quando seu objetivo é identificar os caminhos que levam a um resultado. Por exemplo, você pode usar o nPath para analisar:

Dados de cliques em sites, para identificar caminhos que levam a vendas acima de um valor especificado

Dados de sensores de processos industriais, para identificar caminhos para a baixa qualidade do produto

Registros de saúde de pacientes individuais, para identificar caminhos que indicam que os pacientes estão em risco de desenvolver doenças como doenças cardíacas ou diabetes

Dados financeiros para indivíduos, para identificar caminhos que fornecem informações sobre riscos de crédito ou fraude

Neste exemplo, vamos utilizar o Teradata Vantage para consultar os eventos que os correntistas tomam em seu Internet Banking e que acabam os levando à entrar em contato com o Customer Support. Em um segundo exemplo, vamos identificar dentro de um outro dataset, qual o caminho que correntistas tomam, em suas interações com o banco, até a busca de um Consultor Financeiro para gerir seu patrimônio, no caso, de alto padrão financeiro.

Utilizaremos este Notebook para realizar as consultas ao banco de dados porém, executaremos a visualização desses resultados através do PowerBI.

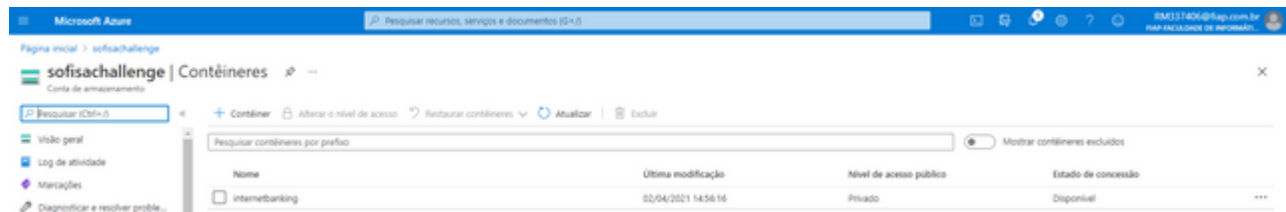
1) Primeramente, vamos realizar nossa conexão ao banco de dados Teradata Vantage.

In [3]:

```
%connect TRDT01
```

Success: 'TRDT01' connection established and activated for user 'nos_usr'

2) Agora, como buscamos dados digitais em Object Store existente no Azure (Blob Storage / ADLS) vamos criar uma Autorização para que nosso usuário tenha acesso à tal Contêiner.



(<https://ibb.co/xHKLNPm>)

In [37]:

```
CREATE AUTHORIZATION DefAuth_AZ
AS DEFINER TRUSTED
USER 'sofisachallenge' /* storage account name */
PASSWORD 'x5Szh3fFXdwCbIEAnMc5ICBkMvu4bEA6B0jbJJ7jU1WTxLwgNxXWYAvQ1VdCzvZMzf0dsyPIInLh3WQpj
jm11bQ==' ; /* storage account key */
```

Out[37]:

Success: 0 rows affected

3) Vamos criar uma Foreign Table, que utilizando a autorização criada acima, irá buscar dados CSV, existentes dentro de nosso Container. O Teradata Vantage possibilita a leitura de arquivos CSV, JSON e PARQUET existentes em um Object Storage.

In [54]:

```
CREATE MULTISET FOREIGN TABLE ft_bank_web_clicks ,FALLBACK ,EXTERNAL SECURITY DEFINER TRUSTED DefAuth_AZ,
    MAP = TD_MAP1
( Location VARCHAR(2048) CHARACTER SET UNICODE CASESPECIFIC,
  Payload DATASET INLINE LENGTH 64000 STORAGE FORMAT CSV
)
USING
(
    LOCATION ('/AZ/sofisachallenge.blob.core.windows.net/internetbanking/')
)
;
```

Out[54]:

Success: 0 rows affected

4) Vamos analisar duas linhas de payload existente dentro deste Container

In [55]:

```
SELECT TOP 2 payload
FROM ft_bank_web_clicks;
```

Out[55]:

	Payload
1	customer_id,session_id,page,datestamp 529,0,FAQ,17/03/2004 16:38:30.000000
2	customer_id,session_id,page,datestamp 529,0,ACCOUNT SUMMARY,17/03/2004 16:35:59.000000

5) Podemos com a consulta abaixo, listar quais são as CSVKEYS de nosso dataset. Isso nos auxiliará nos próximos passos para tabelar esse CSV

In [56]:

```
SELECT DISTINCT * FROM DATASET_KEYS (ON (SELECT payload FROM
ft_bank_web_clicks)) AS
csvKeys;
```

Out[56]:

	DatasetKeys
1	datestamp
2	session_id
3	page
4	customer_id

5) Neste momento, podemos já executar a função nPath diretamente contra o container listado. Nossa consulta abaixo, irá tabelar os dados do arquivo CSV e, em tempo de execução, utilizar a tabela dentro da função nPath. De maneira simples, o resultado da função será o caminho percorrido pelos correntistas, dentro do Internet Banking, até a página de Customer Support. A coluna "cnt" corresponde a quantidade de vezes que esse caminho foi percorrido pelos diversos usuários do Internet Banking.

In [71]:

```
SELECT path, count(*) as cnt
FROM nPath
(ON (SELECT CAST(payload..customer_id AS INTEGER) customer_id,
CAST(payload..session_id AS INTEGER) session_id,
CAST(payload..page AS VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC) page,
CAST(payload..datestamp AS TIMESTAMP(6) FORMAT 'DD-MM-YYYYBHH:MI:SSDS(F)Z') datestamp
FROM ft_bank_web_clicks)
PARTITION BY customer_id, SESSION_ID
ORDER BY datestamp
USING
Mode (NONOVERLAPPING)
Pattern ('PAGE{1,4}.CSP')
Symbols (TRUE AS PAGE,
page = 'CUSTOMER SUPPORT' AS CSP)
Result (Accumulate(page OF ANY (CSP, page)) as path)
) AS dt GROUP BY path HAVING cnt >= 11 ORDER BY cnt desc;
```

Out[71]:

	path
1	[ACCOUNT SUMMARY, CUSTOMER SUPPORT]
2	[ACCOUNT SUMMARY, ACCOUNT HISTORY, CUSTOMER SUPPORT]
3	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, CUSTOMER SUPPORT]
4	[ACCOUNT SUMMARY, FUNDS TRANSFER, CUSTOMER SUPPORT]
5	[ACCOUNT SUMMARY, FAQ, CUSTOMER SUPPORT]
6	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, CUSTOMER SUPPORT]
7	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, CUSTOMER SUPPORT]
8	[ACCOUNT SUMMARY, PROFILE UPDATE, CUSTOMER SUPPORT]
9	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, ACCOUNT HISTORY, CUSTOMER SUPP
10	[ACCOUNT SUMMARY, FUNDS TRANSFER, ACCOUNT HISTORY, CUSTOMER SUPPORT]
11	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, FUNDS TRANSFER, CUSTOMER SUPP
12	[ACCOUNT SUMMARY, FUNDS TRANSFER, ACCOUNT SUMMARY, CUSTOMER SUPPORT]
13	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, FAQ, CUSTOMER SUPPORT]
14	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, FAQ, CUSTOMER SUPPORT]
15	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, ACCOUNT HISTORY, CUSTOMER SUPPOF
16	[ACCOUNT SUMMARY, FAQ, ONLINE STATEMENT ENROLLMENT, CUSTOMER SUPPORT]
17	[ACCOUNT SUMMARY, FUNDS TRANSFER, FUNDS TRANSFER, CUSTOMER SUPPORT]
18	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, ACCOUNT SUMMARY, CUSTOMER SUP
19	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, FUNDS TRANSFER, CUSTOMER SUPPORT]
20	[ACCOUNT SUMMARY, ACCOUNT HISTORY, FUNDS TRANSFER, CUSTOMER SUPPORT]
21	[ACCOUNT SUMMARY, FAQ, PROFILE UPDATE, CUSTOMER SUPPORT]
22	[ACCOUNT SUMMARY, ACCOUNT HISTORY, ACCOUNT HISTORY, CUSTOMER SUPPORT]
23	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, ACCOUNT HISTORY, CUSTC
24	[ACCOUNT SUMMARY, ACCOUNT HISTORY, ACCOUNT SUMMARY, CUSTOMER SUPPOF
25	[ACCOUNT SUMMARY, ACCOUNT HISTORY, FAQ, CUSTOMER SUPPORT]
26	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, PROFILE UPDATE, CUSTOM
27	[ACCOUNT SUMMARY, FUNDS TRANSFER, FAQ, CUSTOMER SUPPORT]
28	[ACCOUNT SUMMARY, FAQ, ACCOUNT HISTORY, CUSTOMER SUPPORT]

6) Se for de nosso interesse, podemos materializar a tabela antes de executar a função nPath. Veja que o resultado da função é o mesmo.

In [61]:

```
CREATE MULTISET TABLE bank_web_clicks AS (  
  SELECT CAST(payload..customer_id AS INTEGER) customer_id,  
    CAST(payload..session_id AS INTEGER) session_id,  
    CAST(payload..page AS VARCHAR(100) CHARACTER SET LATIN NOT CASESPECIFIC) page,  
    CAST(payload..datestamp AS TIMESTAMP(6) FORMAT 'DD-MM-YYYYBHH:MI:SSDS(F)Z') datestamp  
  FROM ft_bank_web_clicks  
)  
WITH DATA  
PRIMARY INDEX ( customer_id );
```

Out[61]:

Success: 0 rows affected

In [72]:

```
SELECT path, count(*) as cnt
FROM nPath
(ON bank_web_clicks
PARTITION BY customer_id, SESSION_ID
ORDER BY datestamp
USING
Mode (NONOVERLAPPING)
Pattern ('PAGE{1,4}.CSP')
Symbols (TRUE AS PAGE,
page = 'CUSTOMER SUPPORT' AS CSP)
Result (Accumulate(page OF ANY (CSP, page)) as path)
) AS dt GROUP BY path HAVING cnt >= 11 ORDER BY cnt desc;
```

Out[72]:

	path
1	[ACCOUNT SUMMARY, CUSTOMER SUPPORT]
2	[ACCOUNT SUMMARY, ACCOUNT HISTORY, CUSTOMER SUPPORT]
3	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, CUSTOMER SUPPORT]
4	[ACCOUNT SUMMARY, FUNDS TRANSFER, CUSTOMER SUPPORT]
5	[ACCOUNT SUMMARY, FAQ, CUSTOMER SUPPORT]
6	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, CUSTOMER SUPPORT]
7	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, CUSTOMER SUPPORT]
8	[ACCOUNT SUMMARY, PROFILE UPDATE, CUSTOMER SUPPORT]
9	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, ACCOUNT HISTORY, CUSTOMER SUPP
10	[ACCOUNT SUMMARY, FUNDS TRANSFER, ACCOUNT HISTORY, CUSTOMER SUPPORT]
11	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, FUNDS TRANSFER, CUSTOMER SUPP
12	[ACCOUNT SUMMARY, FUNDS TRANSFER, ACCOUNT SUMMARY, CUSTOMER SUPPORT]
13	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, FAQ, CUSTOMER SUPPORT]
14	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, FAQ, CUSTOMER SUPPORT]
15	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, ACCOUNT HISTORY, CUSTOMER SUPPOF
16	[ACCOUNT SUMMARY, FAQ, ONLINE STATEMENT ENROLLMENT, CUSTOMER SUPPORT]
17	[ACCOUNT SUMMARY, FUNDS TRANSFER, FUNDS TRANSFER, CUSTOMER SUPPORT]
18	[ACCOUNT SUMMARY, VIEW DEPOSIT DETAILS, ACCOUNT SUMMARY, CUSTOMER SUF
19	[ACCOUNT SUMMARY, ACCOUNT SUMMARY, FUNDS TRANSFER, CUSTOMER SUPPORT]
20	[ACCOUNT SUMMARY, ACCOUNT HISTORY, FUNDS TRANSFER, CUSTOMER SUPPORT]
21	[ACCOUNT SUMMARY, FAQ, PROFILE UPDATE, CUSTOMER SUPPORT]
22	[ACCOUNT SUMMARY, ACCOUNT HISTORY, ACCOUNT HISTORY, CUSTOMER SUPPORT]
23	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, ACCOUNT HISTORY, CUSTO
24	[ACCOUNT SUMMARY, ACCOUNT HISTORY, ACCOUNT SUMMARY, CUSTOMER SUPPOF
25	[ACCOUNT SUMMARY, ACCOUNT HISTORY, FAQ, CUSTOMER SUPPORT]
26	[ACCOUNT SUMMARY, ONLINE STATEMENT ENROLLMENT, PROFILE UPDATE, CUSTOM
27	[ACCOUNT SUMMARY, FUNDS TRANSFER, FAQ, CUSTOMER SUPPORT]
28	[ACCOUNT SUMMARY, FAQ, ACCOUNT HISTORY, CUSTOMER SUPPORT]

In []:

No exemplo abaixo, vamos executar a função nPath contra um dataset maior. Neste caso, desejamos encontrar qual o caminho que correntistas tomam, em suas interações com o banco, até a busca de um Consultor Financeiro para gerir seu patrimônio, no caso, de alto padrão financeiro.

Vamos analisar um amostra de 10 linhas deste dataset

In [30]:

```
SELECT customer_identifier, interaction_timestamp, interaction_type, product_category, interaction_type || '_' || product_category AS event,
       marketing_category, marketing_description, marketing_placement, sales_channel,
       conversion_sales, conversion_cost, conversion_margin
FROM NOS_USR.ich_banking SAMPLE 10;
```

Out[30]:

	customer_identifier	interaction_timestamp	interaction_type
1	695117528xdmrpxt390016463	2016-02-26 18:47:41.000000-07:00	BROWSE
2	705162011ucdhowq550823887	2016-02-09 23:51:43.000000-07:00	BROWSE
3	313591543ljyzmub185932883	2016-03-12 18:28:06.000000-07:00	CLICK
4	822096043pjeonlj902248054	2016-01-05 18:59:48.000000-07:00	CLICK
5	503474559ettjrhh914843574	2016-02-09 04:13:58.000000-07:00	COMPARE
6	448535952nsscqw660956167	2016-01-15 17:01:10.000000-07:00	BROWSE
7	511411935bsfamlg851955772	2016-03-14 16:02:15.000000-07:00	COMPLETE_APPLICATION
8	287707283phuhdqn646619015	2016-02-09 17:28:13.000000-07:00	BROWSE
9	214906976tdhjgpd178495968	2016-03-25 15:09:36.000000-07:00	COMPARE
10	945479006wrcosbv969956658	2016-03-25 15:00:37.000000-07:00	BROWSE

Agora vamos executar a função nPATH e analisar o resultado.

In [42]:

```
SELECT interaction_type_list,count(*) FROM nPath (
    ON (
        SELECT customer_identifier, interaction_timestamp, interaction_type, product_c
        ategory, interaction_type || '_' || product_category AS event,
            marketing_category, marketing_description, marketing_placement, sales_chan
        nel,
            conversion_sales, conversion_cost, conversion_margin
        FROM NOS_USR.ich_banking
        WHERE
            product_category <> '-1'
            AND interaction_type || '_' || product_category <> 'STARTS_APPLICATION_WEA
LTH MANAGEMENT'
            AND interaction_type || '_' || product_category <> 'COMPLETE_APPLICATION_W
EALTH MANAGEMENT'
    )
    PARTITION BY customer_identifier
    ORDER BY interaction_timestamp
    USING
    MODE (NONOVERLAPPING)
    -- Limit to a depth of 4
    PATTERN ('(EVENT){4}.ADOPTION')
    SYMBOLS (
        event NOT LIKE 'ACCOUNT_BOOKED%_WEALTH MANAGEMENT' AS EVENT,
        event LIKE 'ACCOUNT_BOOKED%' AND product_category = 'WEALTH MANAGEMENT' AS ADO
PTION
    )
    RESULT (
        ACCUMULATE( event OF ANY(EVENT,ADOPTION) ) AS interaction_type_list
    )
)a group by 1 order by 2 desc;
```


Out[42]:

1	[COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, BROWSE_WEALTH M
2	[COMPLETE_APPLICATION_SAVINGS, ACCOUNT_BOOKED_ONLINE_SAVINGS, COMPAI
3	[COMPLETE_APPLICATION_BROKERAGE, ACCOUNT_BOOKED_ONLINE_BROKERAGE,
4	[COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, ADD_DIRECT_DEPO
5	[COMPLETE_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_O
6	[COMPLETE_APPLICATION_BROKERAGE, ACCOUNT_BOOKED_ONLINE_BROKERAGE,
7	[COMPLETE_APPLICATION_SAVINGS, ACCOUNT_BOOKED_ONLINE_SAVINGS, BROWS
8	[COMPLETE_APPLICATION_BROKERAGE, COMPLETE_APPLICATION_BROKERAGE, AC
9	[STARTS_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLI
10	[COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, BROWSE_WEALTH M
11	[COMPARE_SAVINGS, ACCOUNT_BOOKED_ONLINE_SAVINGS, COMPARE_WEALTH MA
12	[COMPLETE_APPLICATION_BROKERAGE, ACCOUNT_BOOKED_ONLINE_BROKERAGE,
13	[STARTS_APPLICATION_BROKERAGE, COMPLETE_APPLICATION_BROKERAGE, ACCO
14	[COMPARE_CD, ACCOUNT_BOOKED_ONLINE_CD, BROWSE_WEALTH MANAGEMENT, (
15	[COMPLETE_APPLICATION_CD, COMPARE_CD, ACCOUNT_BOOKED_ONLINE_CD, BRO
16	[STARTS_APPLICATION_CD, COMPLETE_APPLICATION_CD, COMPLETE_APPLICATION_
17	[BROWSE_CD, STARTS_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOUNT_I
18	[COMPARE_CD, STARTS_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOUNT_
19	[BROWSE_CD, COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, BROV
20	[BROWSE_BROKERAGE, STARTS_APPLICATION_BROKERAGE, COMPLETE_APPLICATIONI
21	[STARTS_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLI
22	[STARTS_APPLICATION_CD, COMPARE_CD, COMPLETE_APPLICATION_CD, ACCOUNT_
23	[COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, ACCOUNT_BOOKED
24	[COMPARE_CD, COMPLETE_APPLICATION_CD, ACCOUNT_BOOKED_ONLINE_CD, BRO
25	[COMPARE_CD, COMPLETE_APPLICATION_CD, COMPLETE_APPLICATION_CD, ACCOU
26	[COMPLETE_APPLICATION_BROKERAGE, COMPLETE_APPLICATION_BROKERAGE, AC
27	[ACCOUNT_BOOKED_ONLINE_SAVINGS, COMPARE_WEALTH MANAGEMENT, ENROLL_
28	[STARTS_APPLICATION_BROKERAGE, COMPLETE_APPLICATION_BROKERAGE, COMP

Power BI

Vamos utilizar os dois exemplos acima e nos conectar ao banco através do PowerBI e executar as funções através dele!