

# CHECKPOINT 3

**Projeto ETL – Dimensão Cliente por Faixa Etária (DBurger – Parte II)**

**Professor:** Prof. Salvio Padlipskas

**Turma:** 2TSCPR

**Grupo Insight Hunters:**

- Diego Alves Moreira – RM552603
- Guilherme Yuiti Matsushita Nakamura – RM85355
- João Pedro de Souza Nunes – RM: 554066

**SÃO PAULO, 2025**

## Sumário

<b>1) INTRODUÇÃO .....</b>	<b>3</b>
<b>2) PREPARAÇÃO DO AMBIENTE .....</b>	<b>3</b>
<b>3) DESENVOLVIMENTO .....</b>	<b>6</b>
a) Criação do Data Factory e Conexões .....	6
b) Criando os Datasets .....	8
<b>4) 1ª ENTREGA .....</b>	<b>11</b>
<b>5) 2ª ENTREGA .....</b>	<b>12</b>
a) Populado as Dimensões .....	12
b) Fato Venda .....	13
<b>6) 3ª ENTREGA .....</b>	<b>15</b>
a) VB_DIM_CLIENTE_INSIGHTHUNTERS .....	15
b) VB_DIM_FAIXA_ETARIA_INSIGHTHUNTERS .....	16
c) VB_FTO_VENDA_INSIGHTHUNTERS .....	17

## 1) INTRODUÇÃO

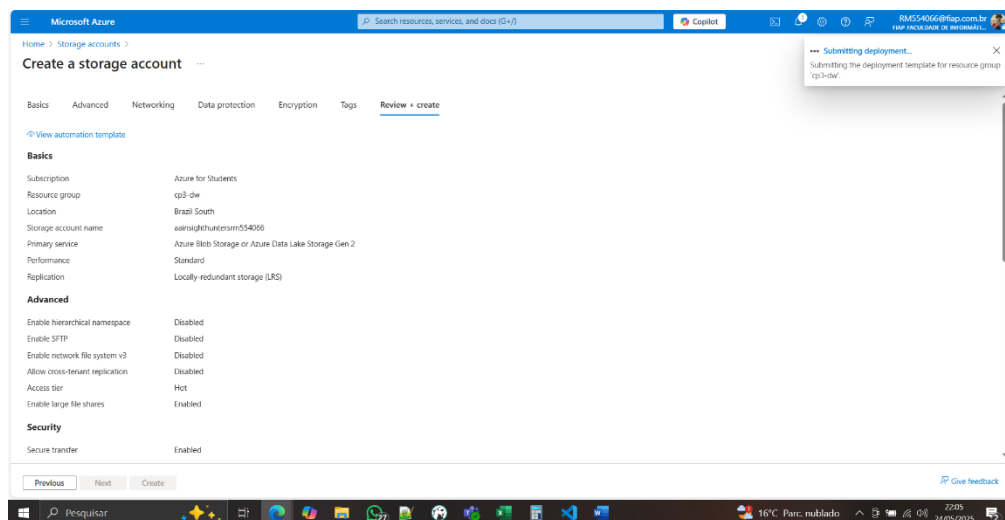
Este trabalho tem como objetivo gerar um pipeline ETL robusto que:

- Calcule a idade dos clientes a partir da data de nascimento.
- Classifique cada cliente em uma faixa etária definida.
- Associe estratégias comerciais adequadas a cada faixa.
- Carregue os dados transformados em tabelas SQL para posterior análise e segmentação.

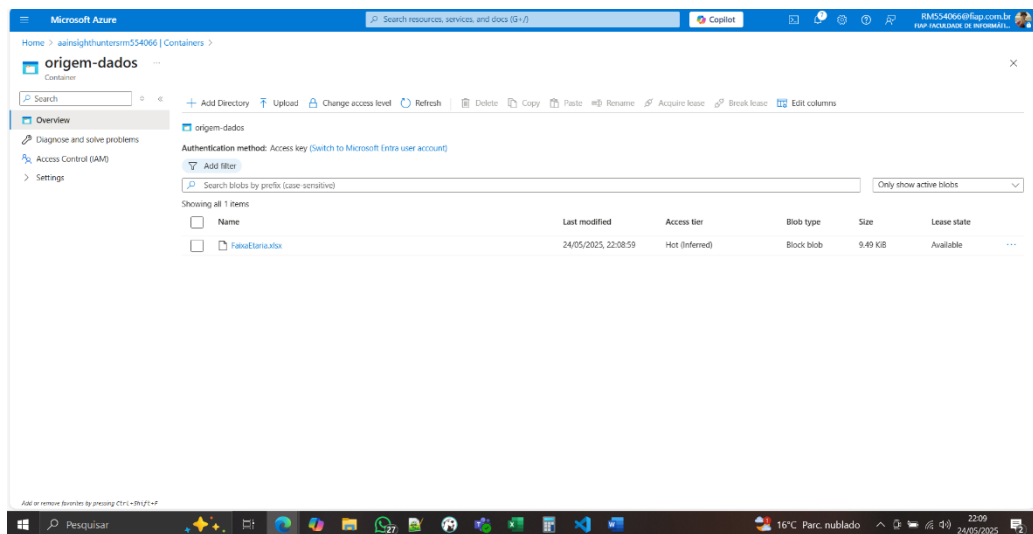
## 2) PREPARAÇÃO DO AMBIENTE

Primeiro preparamos o ambiente foi necessário criar as ferramentas em cloud que utilizaríamos no Azure Data Factory

### Storage Account



## Enviando os dados de origem



## Criando SQL Server via Script

```
--          SCRIPT PARA CRIAÇÃO DO SQL SERVER E BANCO SQL

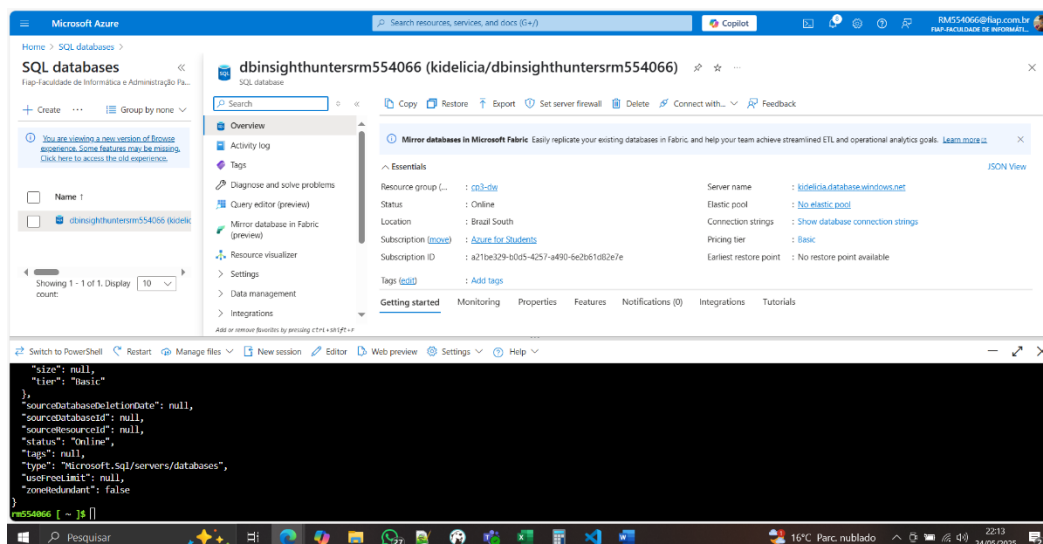
-- Cria o SGBD Azure SQL chamado sqlserver-kidelicia
az sql server create -l brazilsouth -g cp3-dw -n kidelicia -u admsql -p
db@FIAP25 --enable-public-network true

-- Cria o banco de dados chamado dbinsighthuntersrm554066
az sql db create -g cp3-dw -s kidelicia -n dbinsighthuntersrm554066 --
service-objective Basic --backup-storage-redundancy Local --zone-
redundant false

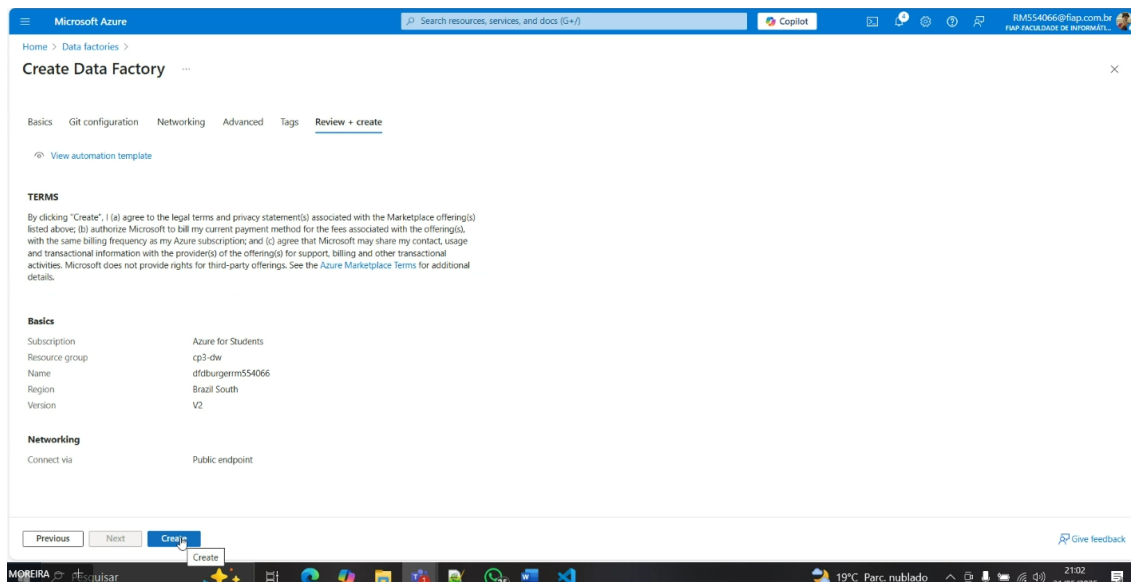
-- Libera o acesso para qualquer endereço IP da rede (somente para testes
iniciais)
az sql server firewall-rule create -g cp3-dw -s kidelicia -n AllowAll --
start-ip-address 0.0.0.0 --end-ip-address 255.255.255.255

-- Usuário e senha do banco de dados
```

```
-- admsql
-- db@FIAP25
```



## Criando Data Factory

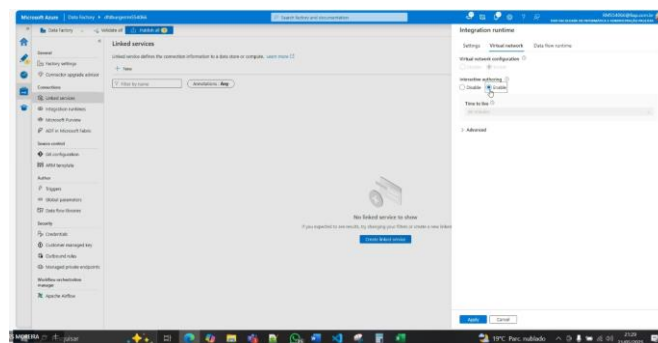


### 3) DESENVOLVIMENTO

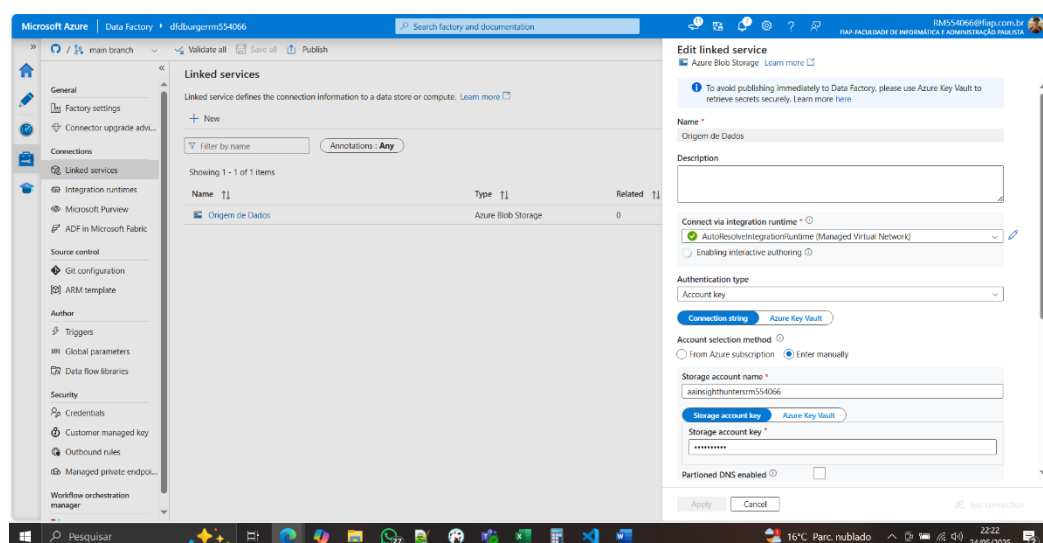
#### a) Criação do Data Factory e Conexões

O *Data Factory* foi configurado para orquestrar os *pipelines*. A conexão com o *Storage Account* e o SQL Server garantiu a integração entre fontes de dados.

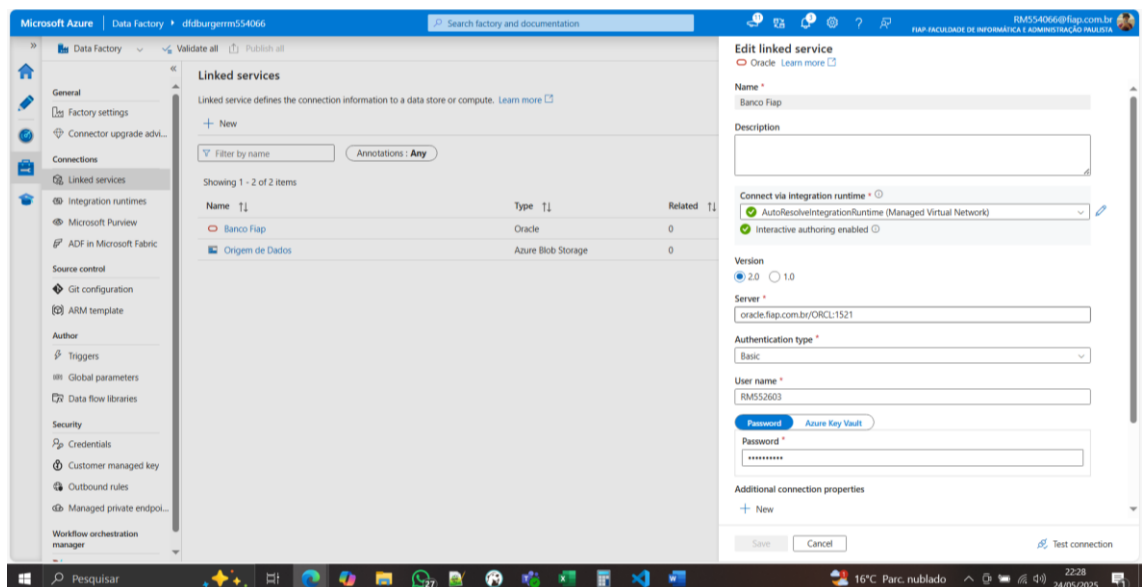
Em todos os passos habilitamos a virtualização para facilidade de ver os dados



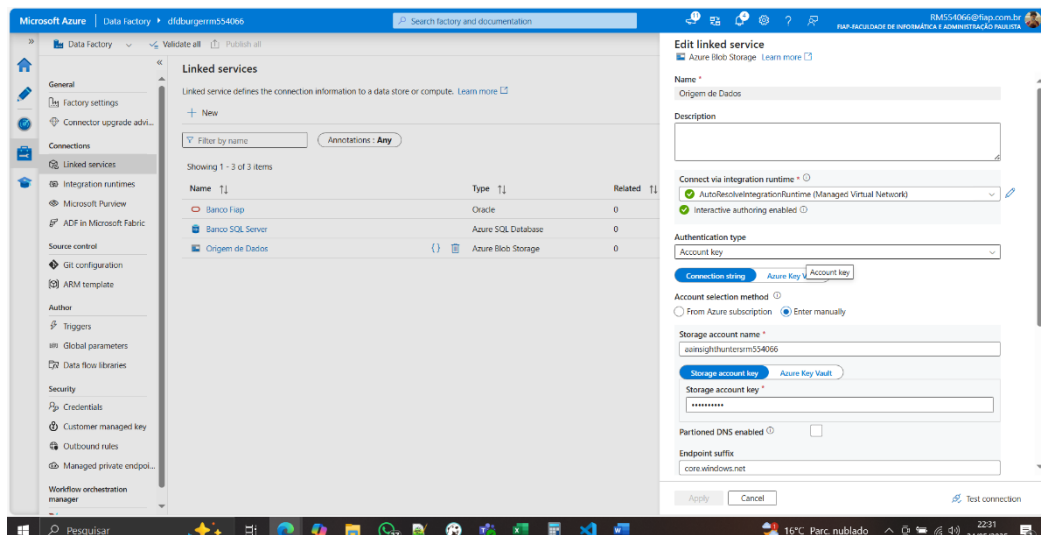
Storage Account



## ORACLE (Banco Fiap)



## SQL Server

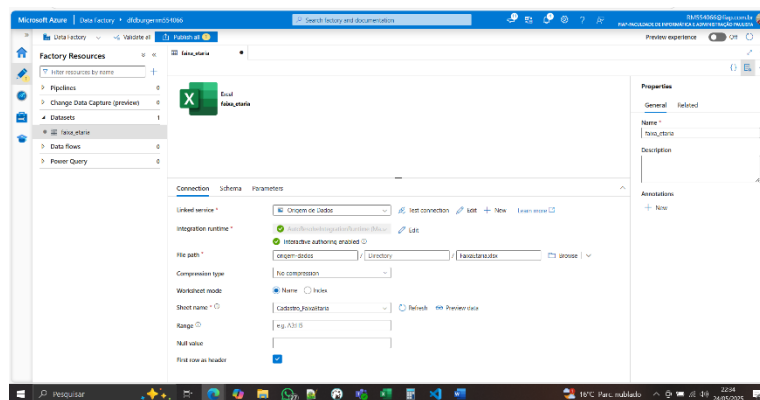


## b) Criando os Datasets

Para armazenar os dados brutos, foi criado um *Storage Account* no Azure, contendo dois containers: um para dados de entrada (*input*) e outro para saída (*output*).

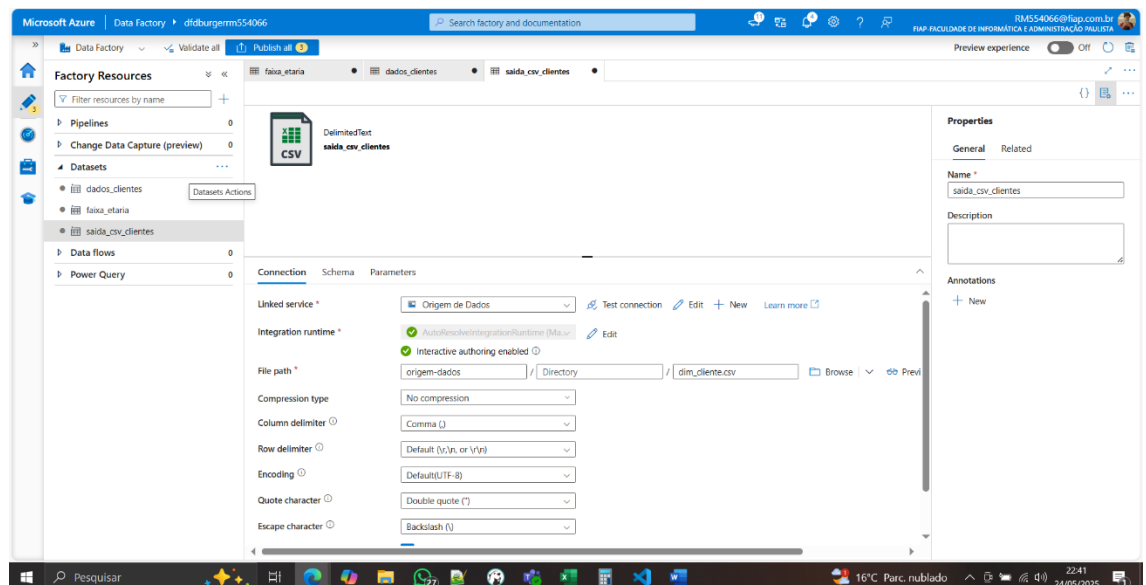
### Dados de entrada (*input*):

FaixaEtaria.xlsx



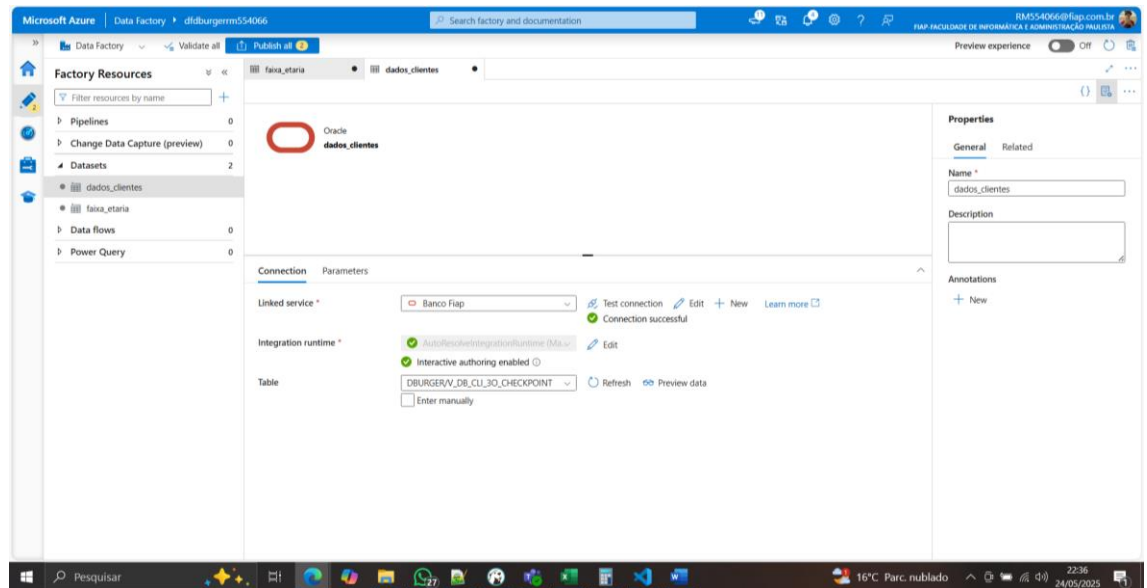
### Dados de saída (*output*):

dim\_cliente.csv

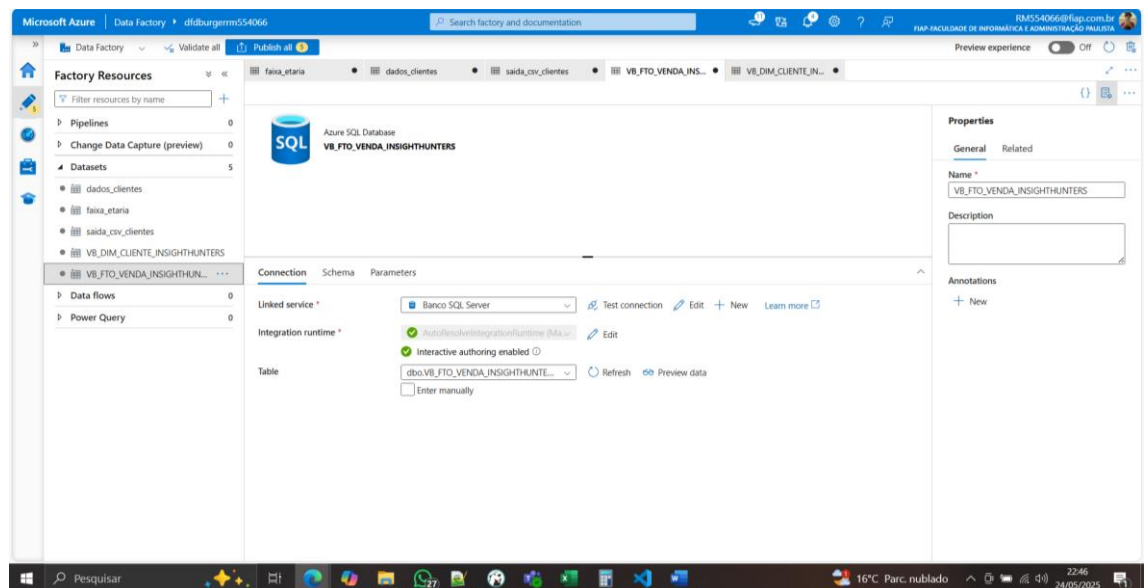




## DBURGER.V\_DB\_CLI\_30\_CHECKPOINT



## VB\_FTO\_VENDA\_INSIGHTHUNTERS



## VB\_DIM\_CLIENTE\_INSIGHTHUNTERS

The screenshot displays the Microsoft Azure Data Factory portal interface. The left-hand navigation pane shows the 'Factory Resources' tree, with 'Datasets' expanded to reveal several datasets, including 'VB\_DIM\_CLIENTE\_INSIGHTHUNTERS'. The main workspace is titled 'VB\_DIM\_CLIENTE\_INSIGHTHUNTERS' and shows the 'Connection' tab. The 'Linked service' is set to 'Banco SQL Server'. The 'Integration runtime' is 'AutoResolveIntegrationRuntime (Managed)'. The 'Table' is 'dbo.VB\_DIM\_CLIENTE\_INSIGHTHUNTERS'. The 'Properties' pane on the right shows the 'Name' as 'VB\_DIM\_CLIENTE\_INSIGHTHUNTERS' and a blank 'Description' field. The bottom of the screen shows the Windows taskbar with the system clock at 22:46 on 24/05/2025.

## VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS

The screenshot displays the Microsoft Azure Data Factory portal interface. The left-hand navigation pane shows the 'Factory Resources' tree, with 'Datasets' expanded to reveal several datasets, including 'VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS'. The main workspace is titled 'VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS' and shows the 'Connection' tab. The 'Linked service' is set to 'Banco SQL Server'. The 'Integration runtime' is 'AutoResolveIntegrationRuntime (Managed)'. The 'Table' is 'dbo.VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS'. The 'Properties' pane on the right shows the 'Name' as 'VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS' and a blank 'Description' field. The bottom of the screen shows the Windows taskbar with the system clock at 22:47 on 24/05/2025.

#### 4) 1ª ENTREGA

Primeiro criamos uma *Pipelines* para poder receber os dados da oracle executado a seguinte query:

```
SELECT * FROM DBURGER.V_DB_CLI_30_CHECKPOINT
WHERE SG_ESTADO = 'SP';
```

#### Fluxo Data Fatory

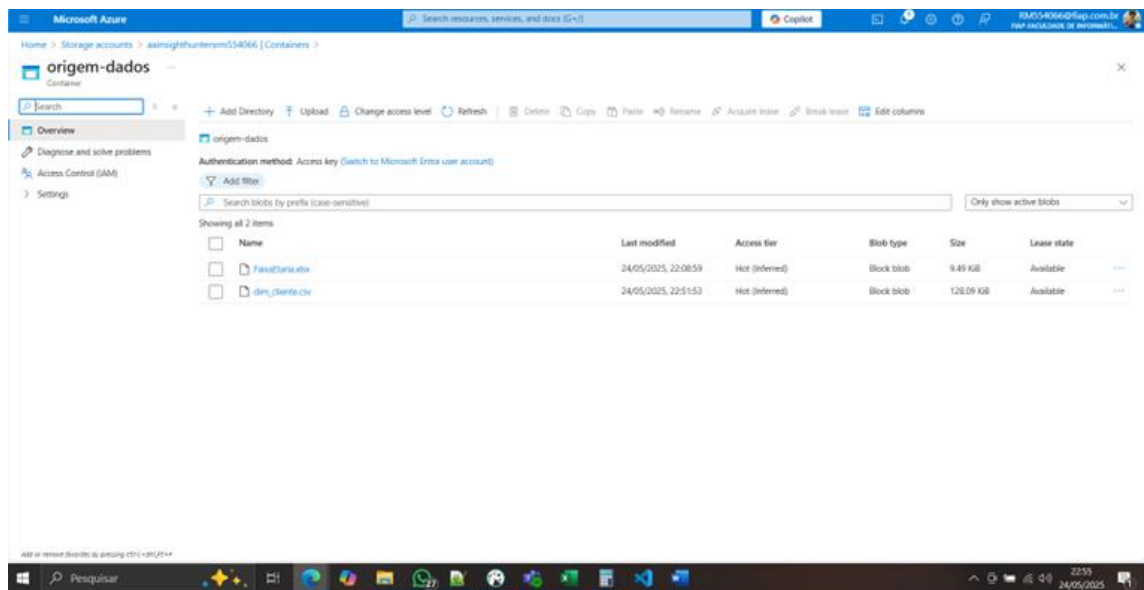
The screenshot shows the Microsoft Azure Data Factory portal. On the left, the 'Factory Resources' pane lists pipelines (Entrega1, Entrega2, Entrega3), datasets, and data flows. The 'Activities' pane shows a list of activities including Move and transform, Synapse, Azure Data Explorer, Azure Function, Batch Service, Data Lake Analytics, General, HDInsight, Iteration & conditionals, Machine Learning, and Power Query. The main canvas displays a pipeline named 'Entrega3' with a single activity 'Copy data' connected to a 'Cifar CSV' dataset. The 'Pipeline status' is 'Succeeded'. Below the pipeline, a table shows the activity details:

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...
Cifar CSV	Succeeded	Copy data	5/25/2025, 4:47:16 AM	57s	AutofrescveintegrationRuntime (East US)	

#### Verificação de dados

The screenshot shows the 'Details' view for a pipeline run in Microsoft Azure Data Factory. The 'Activity run id' is 'd66763d2-90a3-4578-afa7-f28ecbfa797'. The activity is 'Copy data' and it has succeeded. The 'Data read' section shows 16,792 KB of data read, 88 rows read, and 1 peak connection. The 'Data written' section shows 11,921 KB of data written, 1 file written, 88 rows written, and 1 peak connection. The 'Copy duration' is 00:00:53 with a throughput of 2,799 KB/s. The 'Start time' is 5/25/2025, 4:47:17 AM. The 'Used DUs' is 4, 'Used parallel copies' is 1, and 'Duration' is 00:00:53. The 'Details' section shows a 'Queue' activity with a 'Time to first byte' of 00:00:01, 'Reading from source' of 00:00:00, and 'Writing to sink' of 00:00:00. The 'Total duration' is 00:00:45. The 'Data consistency verification' is 'Not verified'. The 'How satisfied or dissatisfied are you with the performance of this copy activity?' section is empty.

## Arquivos



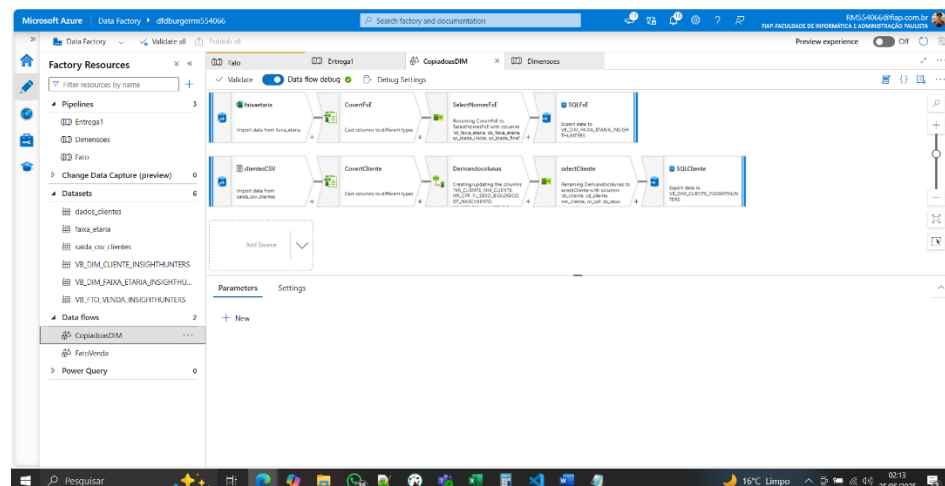
### 5) 2ª ENTREGA

Para segunda entrega executamos o script “cria\_vendebem.sql” os foi adaptado pela nossa equipe para que possa criar o seguinte modelo dimensional:

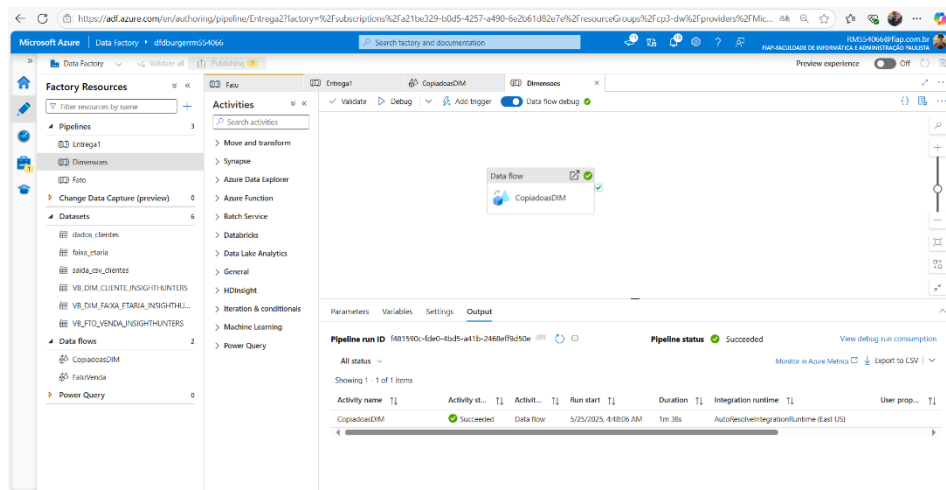
- VB\_DIM\_CLIENTE\_INSIGHTHUNTERS
- VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS
- VB\_FTO\_VENDA\_INSIGHTHUNTERS

#### a) Populado as Dimensões

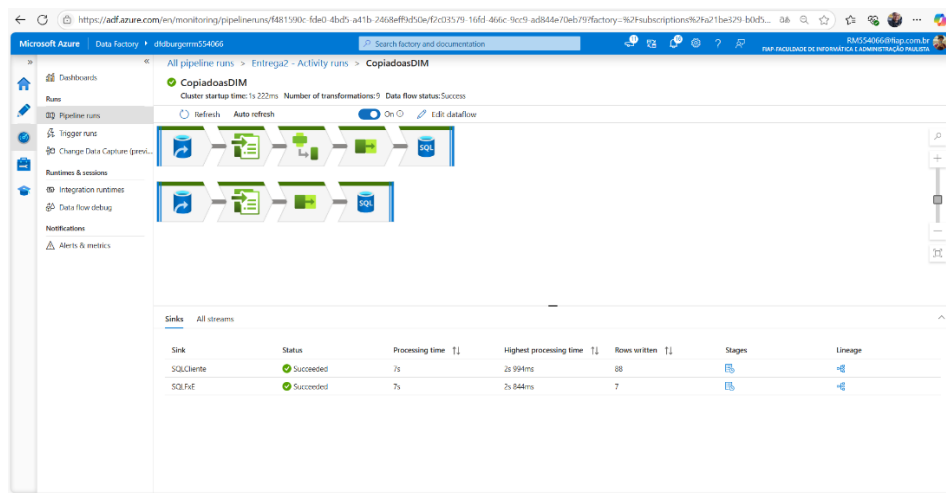
Para receber os dados nas dimensões Cliente e Faixa Etária criamos o seguinte Fluxo:



## Fluxo Data Factory

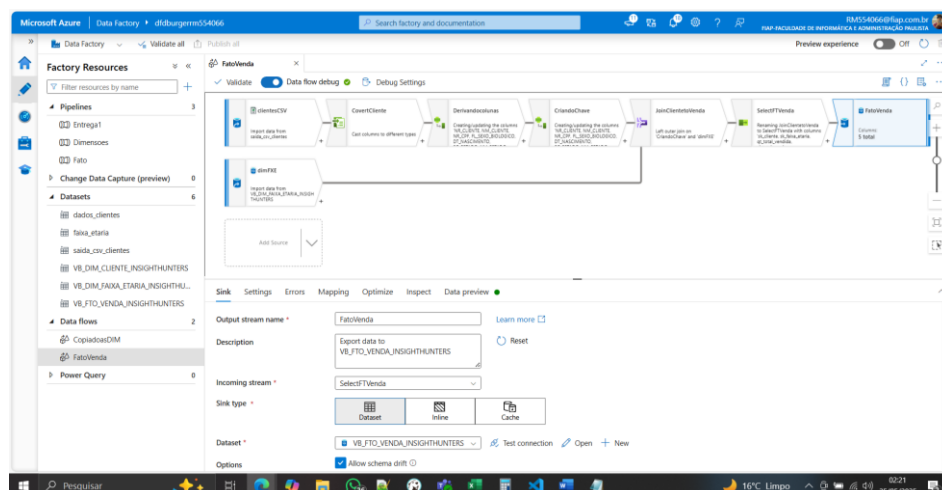


## Verificação de dados

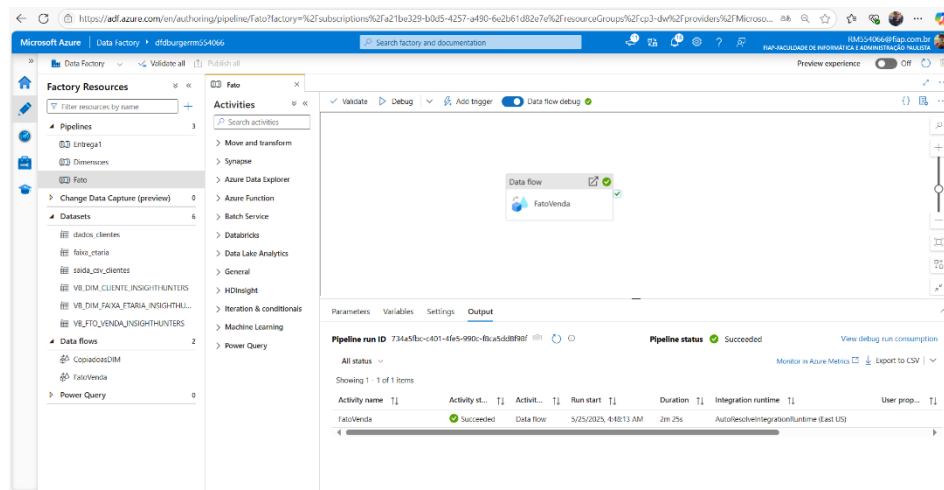


### b) Fato Venda

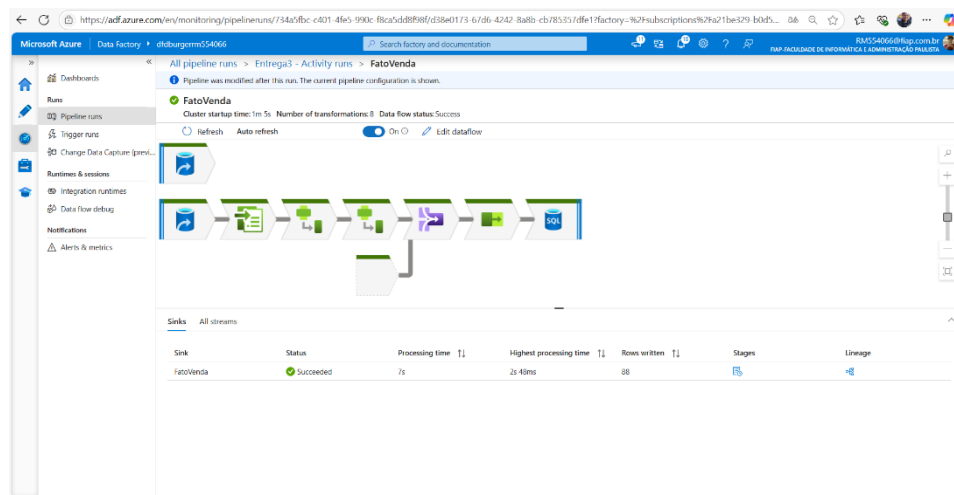
Para finalizar criamos um fluxo de dados para tabela Fato Venda:



## Fluxo Data Fatory



## Verificação de dados



## 6) 3ª ENTREGA

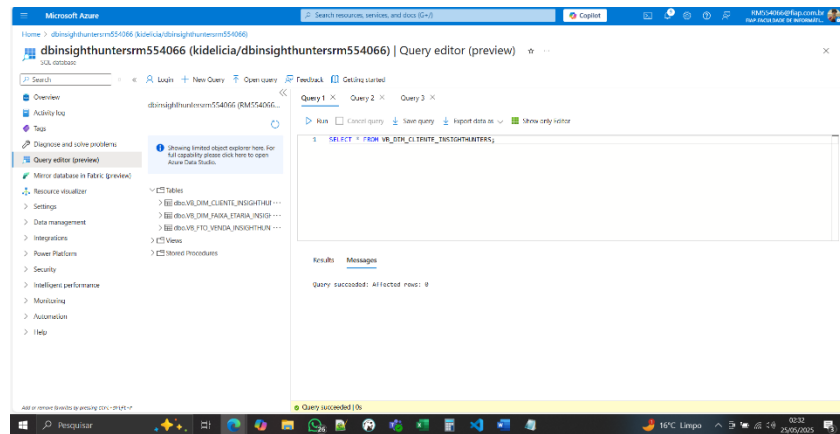
Para validarmos tiramos os *prints* do banco de dados antes e depois de rodar o fluxo de dados:

### a) VB\_DIM\_CLIENTE\_INSIGHTHUNTERS

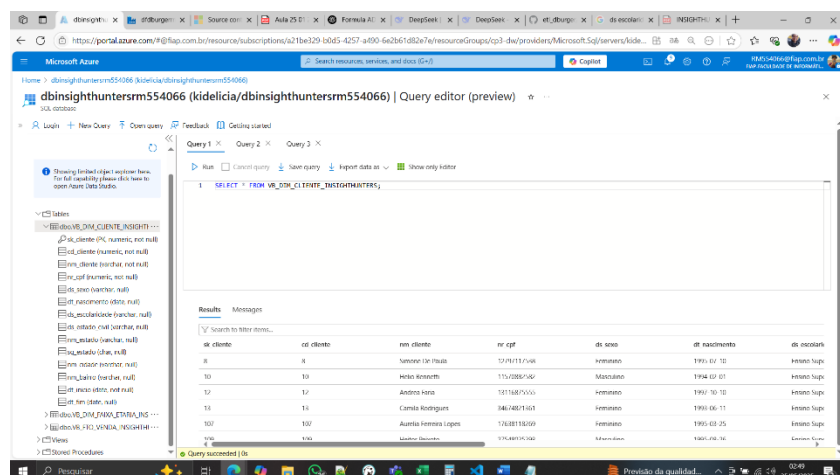
Query:

```
SELECT * FROM VB_DIM_CLIENTE_INSIGHTHUNTERS;
```

Antes:



Depois:

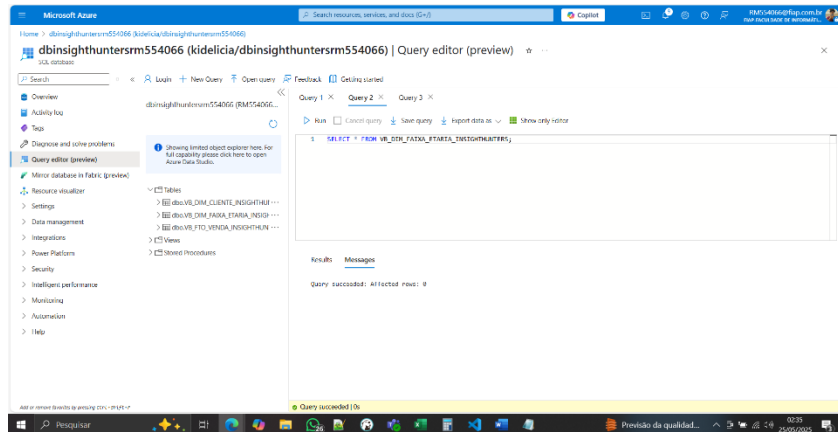


**b) VB\_DIM\_FAIXA\_ETARIA\_INSIGHTHUNTERS**

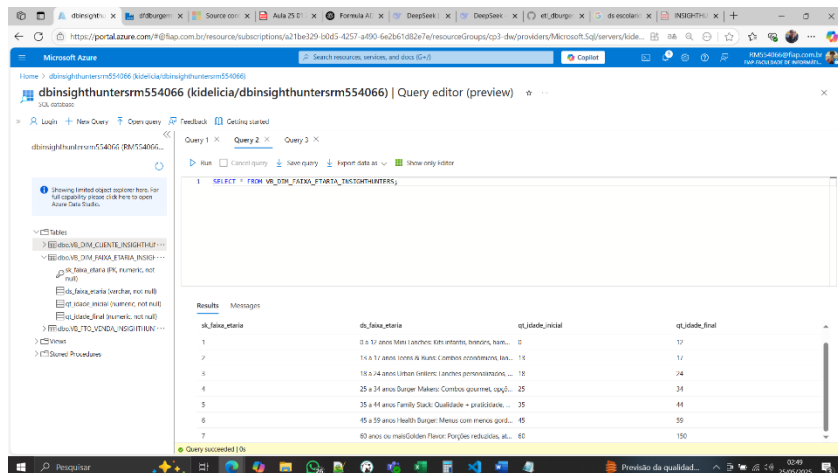
Query:

```
SELECT * FROM VB_DIM_FAIXA_ETARIA_INSIGHTHUNTERS;
```

Antes:



Depois:



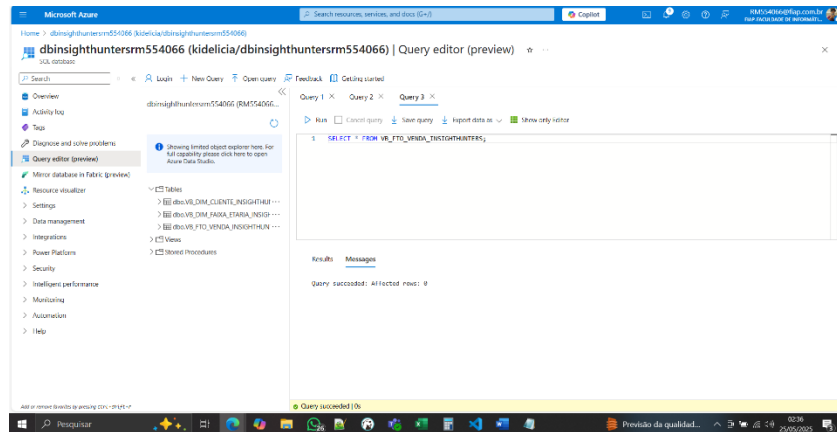


## c) VB\_FTO\_VENDA\_INSIGHTHUNTERS

Query:

```
SELECT * FROM VB_FTO_VENDA_INSIGHTHUNTERS;
```

Antes:



Depois:

