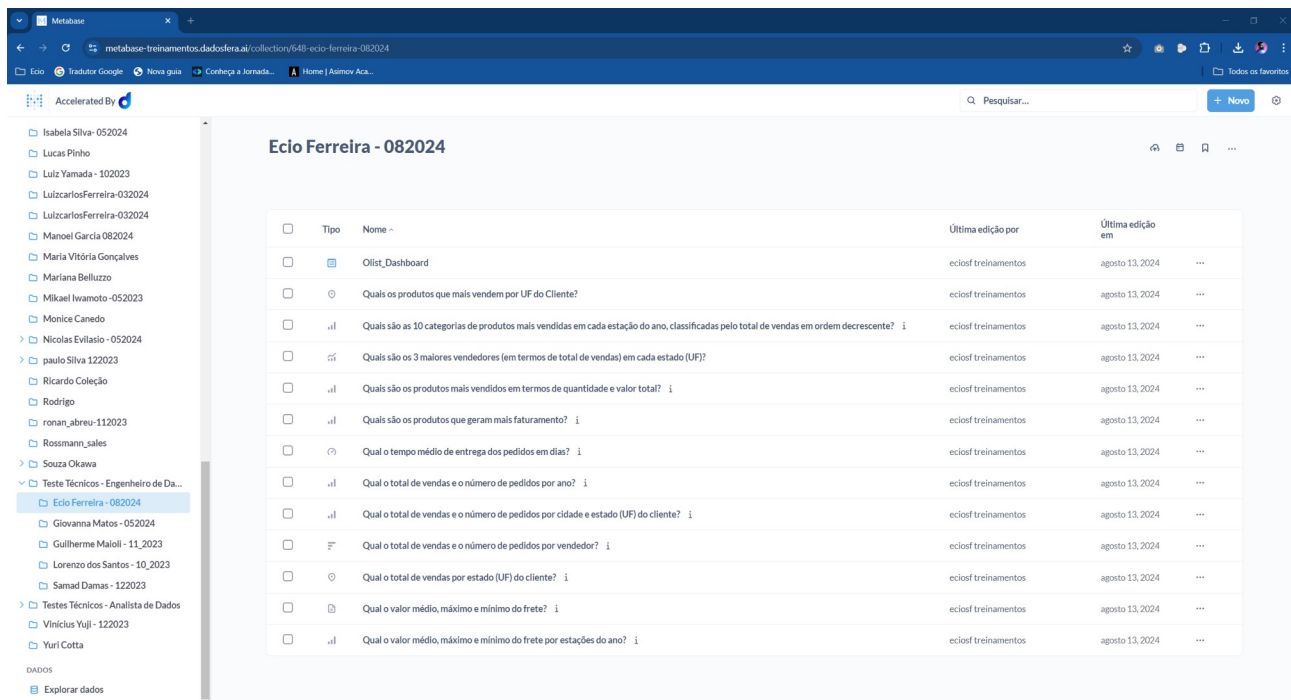


## Item 7 - Sobre Análise de Dados - Analisar

Segue url do painel desenvolvido no Metabase <https://metabase-treinamentos.dadosfera.ai/dashboard/159-olist-dashboard?tab=7-an%C3%A1lise-geral>

No repositório do github existe um diretório com um atalho para o dashboard e dois arquivos PDF com o print das duas abas que compõe esse painel.

Foi criada uma coleção com meu nome Ecio Ferreira – 082024 disponível em <https://metabase-treinamentos.dadosfera.ai/collection/648-ecio-ferreira-082024>



The screenshot shows the Metabase web application. On the left is a sidebar with a tree view of collections. The collection 'Ecio Ferreira - 082024' is selected and highlighted. The main area displays the title 'Ecio Ferreira - 082024' and a table of queries. The table has columns for 'Tipo', 'Nome', 'Última edição por', and 'Última edição em'. The queries listed are related to Olist data analysis, such as 'Olist\_Dashboard', 'Quais os produtos que mais vendem por UF do Cliente?', and 'Quais são os 10 categorias de produtos mais vendidos em cada estação do ano, classificadas pelo total de vendas em ordem decrescente?'.

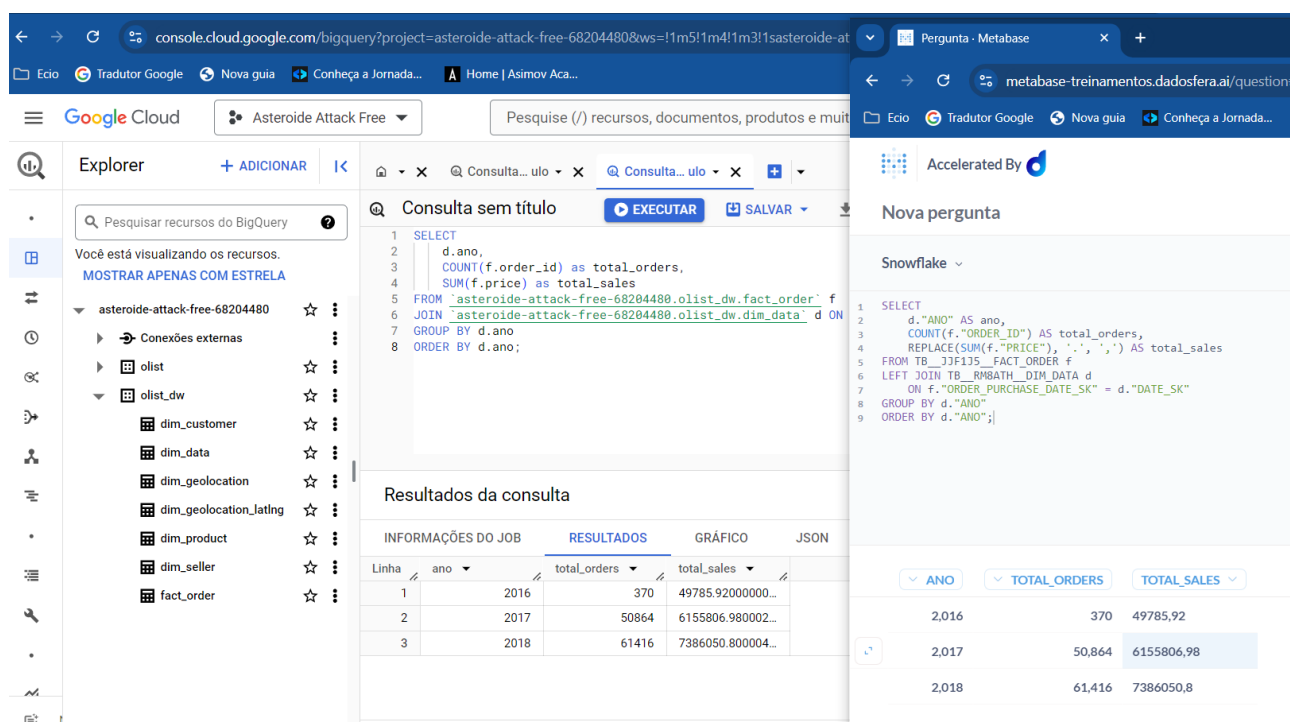
Tipo	Nome	Última edição por	Última edição em
	Olist_Dashboard	eciof treinamentos	agosto 13, 2024
	Quais os produtos que mais vendem por UF do Cliente?	eciof treinamentos	agosto 13, 2024
	Quais são os 10 categorias de produtos mais vendidos em cada estação do ano, classificadas pelo total de vendas em ordem decrescente?	eciof treinamentos	agosto 13, 2024
	Quais são os 3 maiores vendedores (em termos de total de vendas) em cada estado (UF)?	eciof treinamentos	agosto 13, 2024
	Quais são os produtos mais vendidos em termos de quantidade e valor total?	eciof treinamentos	agosto 13, 2024
	Quais são os produtos que geram mais faturamento?	eciof treinamentos	agosto 13, 2024
	Qual o tempo médio de entrega dos pedidos em dias?	eciof treinamentos	agosto 13, 2024
	Qual o total de vendas e o número de pedidos por ano?	eciof treinamentos	agosto 13, 2024
	Qual o total de vendas e o número de pedidos por cidade e estado (UF) do cliente?	eciof treinamentos	agosto 13, 2024
	Qual o total de vendas e o número de pedidos por vendedor?	eciof treinamentos	agosto 13, 2024
	Qual o total de vendas por estado (UF) do cliente?	eciof treinamentos	agosto 13, 2024
	Qual o valor médio, máximo e mínimo do frete?	eciof treinamentos	agosto 13, 2024
	Qual o valor médio, máximo e mínimo do frete por estações do ano?	eciof treinamentos	agosto 13, 2024

A modelagem dimensional e as queries SQL que elaborei anteriormente me proporcionam uma base sólida para a etapa de análise de dados no Metabase. A estrutura dimensional que criei organiza os dados de forma otimizada para consultas analíticas, o que facilitará a exploração e a compreensão das informações durante a análise. As queries SQL que desenvolvi me permitem extrair insights específicos e responder perguntas cruciais sobre o negócio.

O Metabase, com seu suporte a queries personalizadas, potencializa essa etapa de análise. Posso reutilizar as queries SQL que já criei para construir visualizações e painéis

informativos, sem precisar reescrever ou adaptar o código. Essa flexibilidade agilizará meu processo de análise e me permitirá focar na interpretação dos dados e na geração de valor.

Com as tabelas do data warehouse importadas para o Metabase, posso usar minhas queries SQL para criar visualizações como gráficos de barras, linhas, tabelas e mapas. Essas visualizações apresentarão informações relevantes sobre vendas por categoria, evolução temporal, produtos mais vendidos, avaliação média por categoria e distribuição geográfica das vendas.



Como pode ser observado na imagem acima, obtive o mesmo resultado tanto no BigQuery quanto no Metabase. Isso foi possível porque **reaproveitei a lógica desenvolvida na etapa de modelagem para aplicar na etapa de análise**. Dessa forma, consigo até mesmo validar os dados gerados pelo Metabase, garantindo que o resultado apresentado está de acordo com o esperado e com a estrutura do meu modelo de dados.

**O principal ganho dessa abordagem é a consistência e a confiabilidade dos resultados.** Ao utilizar a mesma lógica tanto na construção do data warehouse quanto na análise dos dados no Metabase, evito discrepâncias e erros que poderiam surgir caso as queries fossem reescritas ou adaptadas. Além disso, essa estratégia promove a **eficiência e a agilidade**, pois não preciso reinventar a roda a cada nova análise. Posso me concentrar em explorar os dados e gerar insights, com a certeza de que as informações são precisas e refletem a estrutura do meu modelo.

## Perguntas Respondidas por cada Query e Descrições para o Metabase

### Queries Gerais:

1. **Query:** -- 1. Vendas Totais por Ano:
  - **Pergunta:** Qual o total de vendas e o número de pedidos por ano?
  - **Descrição no Metabase:** "Vendas Totais por Ano"
2. **Query:** -- 2. Vendas Totais por Vendedor:
  - **Pergunta:** Qual o total de vendas e o número de pedidos por vendedor?
  - **Descrição no Metabase:** "Vendas Totais por Vendedor"
3. **Query:** -- 3. Produtos Mais Vendidos:
  - **Pergunta:** Quais são os produtos mais vendidos em termos de quantidade e valor total?
  - **Descrição no Metabase:** "Produtos Mais Vendidos"
4. **Query:** -- 4. Análise de Tempo de Entrega:
  - **Pergunta:** Qual o tempo médio de entrega dos pedidos em dias?
  - **Descrição no Metabase:** "Tempo Médio de Entrega"
5. **Query:** -- 5. Análise de Valor do Frete:
  - **Pergunta:** Qual o valor médio, máximo e mínimo do frete?
  - **Descrição no Metabase:** "Análise de Valor do Frete"

### Análises Geográficas:

1. **Query:** -- 1. Vendas Totais por Região: -- Por cidade do cliente, incluindo a UF:
  - **Pergunta:** Qual o total de vendas e o número de pedidos por cidade e estado (UF) do cliente?
  - **Descrição no Metabase:** "Vendas Totais por Cidade e UF do Cliente"
2. **Query:** -- 1. Vendas Totais por Região: -- Por estado do cliente:
  - **Pergunta:** Qual o total de vendas e o número de pedidos por estado (UF) do cliente?
  - **Descrição no Metabase:** "Vendas Totais por Estado do Cliente"
3. **Query:** -- 2. Desempenho do Vendedor por Região: -- Por cidade do vendedor, incluindo a UF:
  - **Pergunta:** Qual o desempenho dos vendedores (total de vendas e número de pedidos) em cada cidade e estado (UF)?

- **Descrição no Metabase:** "Desempenho do Vendedor por Cidade e UF"
4. **Query:** -- 2. Desempenho do Vendedor por Região: -- Por estado do vendedor:
- **Pergunta:** Qual o desempenho dos vendedores (total de vendas e número de pedidos) em cada estado (UF)?
  - **Descrição no Metabase:** "Desempenho do Vendedor por Estado"
5. **Query:** -- 3. Produtos Mais Vendidos por Região: -- Por cidade do cliente, incluindo a UF:
- **Pergunta:** Quais são os produtos mais vendidos em cada cidade e estado (UF) do cliente?
  - **Descrição no Metabase:** "Produtos Mais Vendidos por Cidade e UF do Cliente"
6. **Query:** -- 3. Produtos Mais Vendidos por Região: -- Por estado do cliente:
- **Pergunta:** Quais são os produtos mais vendidos em cada estado (UF) do cliente?
  - **Descrição no Metabase:** "Produtos Mais Vendidos por Estado do Cliente"

## Análises Gerais:

-- 1. Vendas Totais por Ano:

```
SELECT
  d."ANO" AS ano,
  COUNT(f."ORDER_ID") AS total_orders,
  (REPLACE(SUM(f."PRICE"), ',', '.'):NUMERIC) AS total_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__RM8ATH__DIM_DATA d
  ON f."ORDER_PURCHASE_DATE_SK" = d."DATE_SK"
GROUP BY d."ANO"
ORDER BY d."ANO";
ORDER BY d."ANO";
```

-- 2. Vendas Totais por Vendedor:

```
SELECT
  s."SELLER_ID" AS seller_id,
  COUNT(f."ORDER_ID") AS total_orders,
  CAST(REPLACE(SUM(f."PRICE"), ',', '.') AS NUMERIC) AS total_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__8HMZ94__DIM_SELLER s
  ON f."SELLER_SK" = s."SELLER_SK"
GROUP BY s."SELLER_ID"
HAVING COUNT(f."ORDER_ID") >= 1000
```

ORDER BY total\_orders DESC

-- 3. Produtos Mais Vendidos:

```
SELECT
  p."PRODUCT_ID" AS product_id,
  COUNT(f."ORDER_ID") AS total_orders,
  (REPLACE(SUM(f."PRICE"), ',', '.'):NUMERIC) AS total_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__HKR4H4__DIM_PRODUCT p
  ON f."PRODUCT_SK" = p."PRODUCT_SK"
GROUP BY p."PRODUCT_ID"
ORDER BY total_orders DESC;
```

-- 4. Análise de Tempo de Entrega:

```
SELECT
  AVG(DATEDIFF('day', d_purchase."DATA_COMPLETA",
  d_delivered."DATA_COMPLETA")) AS average_delivery_days
  FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__RM8ATH__DIM_DATA d_purchase
  ON f."ORDER_PURCHASE_DATE_SK" = d_purchase."DATE_SK"
LEFT JOIN TB__RM8ATH__DIM_DATA d_delivered
  ON f."ORDER_DELIVERED_CUSTOMER_DATE_SK" = d_delivered."DATE_SK";
```

-- 5. Análise de Valor do Frete:

```
SELECT
  REPLACE(AVG(f."FREIGHT_VALUE"), ',', '.') AS average_freight_value,
  REPLACE(MAX(f."FREIGHT_VALUE"), ',', '.') AS max_freight_value,
  REPLACE(MIN(f."FREIGHT_VALUE"), ',', '.') AS min_freight_value
FROM TB__JJF1J5__FACT_ORDER f;
```

-- Análise de Valor do Frete por Estações do Ano:

```
SELECT
  d.estacao AS season,
  AVG(TO_NUMBER(REPLACE(f."FREIGHT_VALUE", ',', '.))) AS average_sales,
  MAX(TO_NUMBER(REPLACE(f."FREIGHT_VALUE", ',', '.))) AS max_sales,
  MIN(TO_NUMBER(REPLACE(f."FREIGHT_VALUE", ',', '.))) AS min_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__RM8ATH__DIM_DATA d
  ON f."ORDER_PURCHASE_DATE_SK" = d."DATE_SK"
GROUP BY d.estacao
ORDER BY d.estacao;
```

**-- Análises Geográficas:**

-- 1. Vendas Totais por Região:

-- Por cidade do cliente, incluindo a UF:

```
SELECT
  COALESCE(gc."GEOLOCATION_CITY", 'Unknown City') AS city,
  COALESCE(gc."GEOLOCATION_STATE", 'Unknown State') AS uf,
  COUNT(f."ORDER_ID") AS total_orders,
  (REPLACE(SUM(f."PRICE"), ',', '.'):NUMERIC) AS total_sales
```

```

FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gc
ON f."CUSTOMER_GEOLOCATION_SK" = gc."GEOLOCATION_SK"
GROUP BY city, uf
ORDER BY total_sales DESC;

```

-- Por estado do cliente:

```

SELECT
COALESCE(gc."GEOLOCATION_STATE", 'Unknown State') AS uf,
COUNT(f."ORDER_ID") AS total_orders,
(REPLACE(SUM(f."PRICE"), ',', ':')::NUMERIC) total_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gc
ON f."CUSTOMER_GEOLOCATION_SK" = gc."GEOLOCATION_SK"
GROUP BY uf
ORDER BY total_sales DESC;

```

-- 2. Desempenho do Vendedor por Região:

-- Por cidade do vendedor, incluindo a UF:

```

SELECT city, uf, seller_id, total_orders, total_sales
FROM (
  SELECT
    gs."GEOLOCATION_CITY" as city,
    gs."GEOLOCATION_STATE" AS uf,
    s."SELLER_ID" as seller_id,
    COUNT(f."ORDER_ID") AS total_orders,
    CAST(REPLACE(SUM(f."PRICE"), ',', ':') AS NUMERIC) total_sales,
    ROW_NUMBER() OVER (PARTITION BY gs."GEOLOCATION_CITY" ORDER BY
total_sales DESC) AS seller_rank
  FROM TB__JJF1J5__FACT_ORDER f
  LEFT JOIN TB__8HMZ94__DIM_SELLER s
    ON f."SELLER_SK" = s."SELLER_SK"
  LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gs
    ON f."SELLER_GEOLOCATION_SK" = gs."GEOLOCATION_SK"
  GROUP BY gs."GEOLOCATION_CITY", gs."GEOLOCATION_STATE", s."SELLER_ID"
)
WHERE seller_rank <= 3
ORDER BY city, total_sales DESC;

```

-- Por estado do vendedor:

```

SELECT UF, Vendedor, total_orders, total_sales
FROM (
  SELECT
    COALESCE(gs."GEOLOCATION_STATE", 'Unknown State') as UF,
    s."SELLER_ID" as Vendedor,
    COUNT(f."ORDER_ID") AS total_orders,
    (REPLACE(SUM(f."PRICE"), ',', ':')::NUMERIC) total_sales,
    ROW_NUMBER() OVER (PARTITION BY gs."GEOLOCATION_STATE" ORDER BY
total_sales DESC) AS seller_rank
  FROM TB__JJF1J5__FACT_ORDER f
  LEFT JOIN TB__8HMZ94__DIM_SELLER s
    ON f."SELLER_SK" = s."SELLER_SK"

```

```

LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gs
  ON f."SELLER_GEOLOCATION_SK" = gs."GEOLOCATION_SK"
GROUP BY gs."GEOLOCATION_STATE", s."SELLER_ID"
)
WHERE seller_rank <= 3
ORDER BY UF, total_sales DESC;

```

-- 3. Produtos Mais Vendidos por Região:

-- Por cidade do cliente, incluindo a UF:

```

SELECT
  gc."GEOLOCATION_CITY" AS cidade,
  gc."GEOLOCATION_STATE" AS uf,
  p."PRODUCT_ID" AS product_id,
  COUNT(f."ORDER_ID") AS total_orders,
  (REPLACE(SUM(f."PRICE"), ',', ':')::NUMERIC) AS total_sales -- Corrected
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__HKR4H4__DIM_PRODUCT p
  ON f."PRODUCT_SK" = p."PRODUCT_SK"
LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gc
  ON f."CUSTOMER_GEOLOCATION_SK" = gc."GEOLOCATION_SK"
GROUP BY cidade, uf, product_id
ORDER BY cidade, total_orders DESC;

```

-- Por estado do cliente:

```

SELECT
  gc."GEOLOCATION_STATE" AS uf,
  p."PRODUCT_ID" AS product_id,
  COUNT(f."ORDER_ID") AS total_orders,
  (REPLACE(SUM(f."PRICE"), ',', ':')::NUMERIC) AS total_sales
FROM TB__JJF1J5__FACT_ORDER f
LEFT JOIN TB__HKR4H4__DIM_PRODUCT p
  ON f."PRODUCT_SK" = p."PRODUCT_SK"
LEFT JOIN TB__F6F8US__DIM_GEOLOCATION gc
  ON f."CUSTOMER_GEOLOCATION_SK" = gc."GEOLOCATION_SK"
GROUP BY uf, product_id
ORDER BY uf, total_orders DESC;

```

Categorias mais vendidas por estação do ano.

```

SELECT season, category, total_orders, total_sales
FROM (
  SELECT
    d.estacao AS season,
    p.product_category_name AS category,
    COUNT(f."ORDER_ID") AS total_orders,
    SUM(CAST(REPLACE(f."PRICE", ',', ':') AS NUMERIC)) AS total_sales,
    ROW_NUMBER() OVER (PARTITION BY d.estacao ORDER BY total_sales DESC)
  AS category_rank
  FROM TB__JJF1J5__FACT_ORDER f
  LEFT JOIN TB__RM8ATH__DIM_DATA d
    ON f."ORDER_PURCHASE_DATE_SK" = d."DATE_SK"
  LEFT JOIN TB__HKR4H4__DIM_PRODUCT p

```

```
        ON f."PRODUCT_SK" = p."PRODUCT_SK"  
    GROUP BY d.estacao, p.product_category_name  
)  
WHERE category_rank <= 10  
ORDER BY season;
```