

TÓPICOS AVANÇADOS EM INTELIGÊNCIA ARTIFICIAL II

DW Northwind + Metabase

Consolidação de ERP (B2B) e SaaS (B2C) com dbt e PostgreSQL em um Data Warehouse

Visão do Projeto

O Cenário

Uma empresa opera com dois modelos de negócio distintos:

- ▶ **Northwind (B2B):** Venda de produtos (ERP legado).
- ▶ **Metabase Sample (B2C):** SaaS de assinaturas e produtos.

O Objetivo

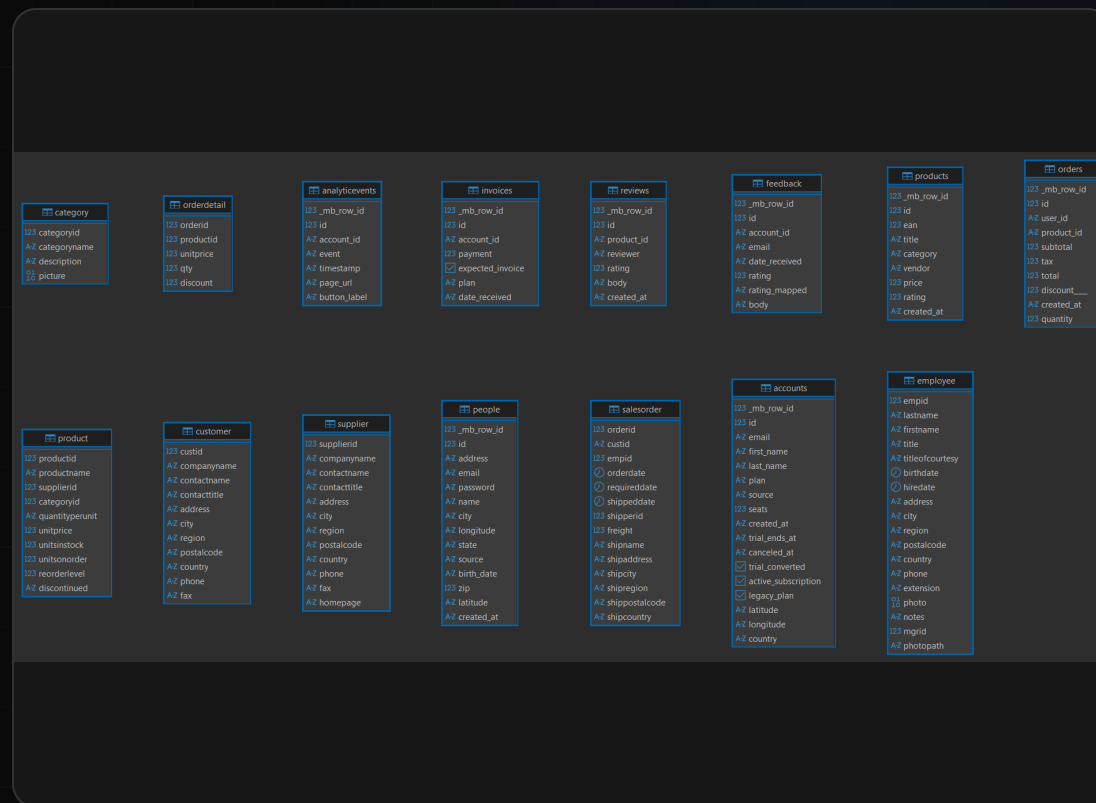
Unificar essas origens díspares em um único **Data Warehouse** no PostgreSQL, utilizando **dbt** para orquestração e transformação, habilitando análises cruzadas de produtos e performance de vendas.

Arquitetura Lógica (FDW)

Unificação via Foreign Data Wrapper

Para evitar pipelines complexos de ingestão inicial, utilizamos o recurso nativo PostgreSQL FDW.

- ▶ As bases northwind e sample_metabase são mapeadas como schemas estrangeiros dentro do DW.
- ▶ O dbt lê diretamente do schema public (Raw), onde essas tabelas estão expostas.



Desafio: Historização (SCD2)

O Problema

Precisávamos historizar mudanças de clientes e produtos (ex: mudança de endereço ou preço), mas as bases de origem **não possuíam colunas confiáveis de auditoria** (`updated_at`).

A Solução: Strategy 'Check'

Adotamos a estratégia de snapshot check do dbt.

- ▶ O dbt cria um hash das colunas monitoradas (ex: `city`, `unit_price`).
- ▶ Quando o hash muda, uma nova versão é criada no DW.
- ▶ **Trade-off:** Aceita-se uma latência de D-1 em troca de não precisar alterar o banco de produção com triggers.

Snapshot Check

Entidade	Colunas Monitoradas
Customers	customer_name, contact_name, first_name, last_name, address, city, region, country, postal_code, plan, segment, seats, active_subscription, canceled_at
Products	product_name, supplier_name, unit_price, rating_score
Suppliers	supplier_name, contact_name, city, region, country
Employees	first_name, last_name, job_title, city, region, country, postal_code, manager_empid

Lógica de Backdating

<div>suppliers_snapshot</div> <div><div>AZ supplier_nk</div><div>AZ source_system</div><div>AZ source_supplier_id</div><div>AZ supplier_name</div><div>AZ contact_name</div><div>AZ contact_title</div><div>AZ city</div><div>AZ region</div><div>AZ country</div><div>AZ phone</div><div>🕒 dw_load_ts</div><div>AZ dbt_scd_id</div><div>🕒 dbt_updated_at</div><div>🕒 dbt_valid_from</div><div>🕒 dbt_valid_to</div></div>	<div>employees_snapshot</div> <div><div>AZ employee_nk</div><div>AZ source_system</div><div>123 source_employee_id</div><div>AZ first_name</div><div>AZ last_name</div><div>AZ job_title</div><div>AZ courtesy_title</div><div>🕒 birth_date</div><div>🕒 hire_date</div><div>AZ city</div><div>AZ region</div><div>AZ country</div><div>AZ postal_code</div><div>AZ phone</div><div>AZ phone_extension</div><div>123 manager_empid</div><div>🕒 dw_load_ts</div><div>AZ dbt_scd_id</div><div>🕒 dbt_updated_at</div><div>🕒 dbt_valid_from</div><div>🕒 dbt_valid_to</div></div>	<div>products_snapshot</div> <div><div>AZ product_nk</div><div>AZ source_system</div><div>123 source_product_id</div><div>AZ supplier_nk</div><div>AZ category_nk</div><div>AZ product_name</div><div>AZ category_name</div><div>AZ supplier_name</div><div>AZ package_details</div><div>123 unit_price</div><div>123 units_in_stock</div><div>123 units_on_order</div><div>123 reorder_level</div><div>123 rating_score</div><div>🕒 product_created_at</div><div><input checked="" type="checkbox"/> is_discontinued</div><div>🕒 dw_load_ts</div><div>AZ dbt_scd_id</div><div>🕒 dbt_updated_at</div><div>🕒 dbt_valid_from</div><div>🕒 dbt_valid_to</div></div>	<div>customers_snapshot</div> <div><div>AZ customer_nk</div><div>AZ source_system</div><div>AZ source_customer_id</div><div>AZ customer_name</div><div>AZ contact_name</div><div>AZ contact_title</div><div>AZ first_name</div><div>AZ last_name</div><div>AZ address</div><div>AZ city</div><div>AZ region</div><div>AZ country</div><div>AZ postal_code</div><div>AZ plan</div><div>AZ segment</div><div>123 seats</div><div><input checked="" type="checkbox"/> trial_converted</div><div><input checked="" type="checkbox"/> active_subscription</div><div><input checked="" type="checkbox"/> legacy_plan</div><div>🕒 created_at</div><div>🕒 trial_ends_at</div><div>🕒 canceled_at</div><div>🕒 dw_load_ts</div><div>AZ dbt_scd_id</div><div>🕒 dbt_updated_at</div><div>🕒 dbt_valid_from</div><div>🕒 dbt_valid_to</div></div>
---	---	---	---

Evitando "Fatos Órfãos"

Como o snapshot começou a rodar hoje, vendas de 1998 ficariam sem dimensão correspondente.

Implementação:

- ▶ Na dimensão, se for a primeira versão do registro, forçamos o `dbt_valid_from` para 1900-01-01.
- ▶ Isso garante que joins históricos (ex: `fact_sales` join `dim_customer`) funcionem para dados legados.

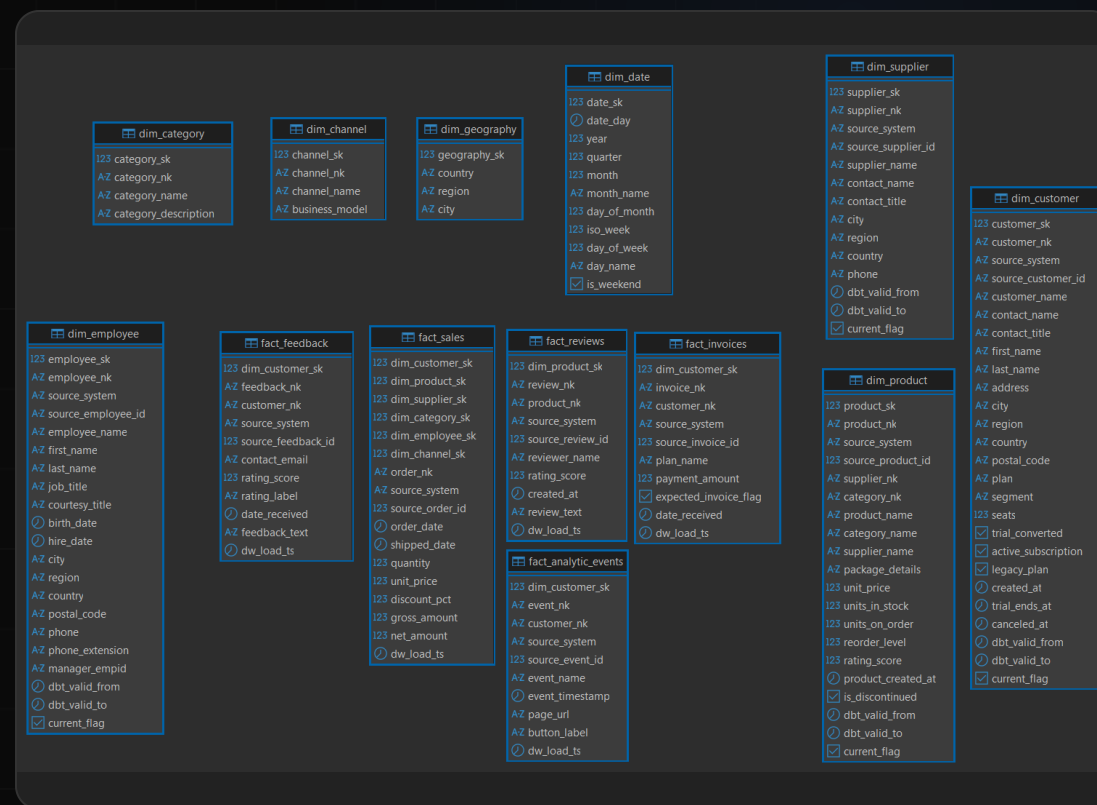
Modelagem Star Schema (DW Core)

Dimensões Unificadas

- ▶ **dim_customer:** Unifica clientes B2B (Northwind) e usuários B2C (Metabase).
- ▶ **dim_product:** Produtos físicos e planos de assinatura.

Fatos de Negócio

- ▶ **fact_sales:** Vendas consolidadas.
- ▶ **fact_analytic_events:** Eventos de clique/navegação do SaaS.



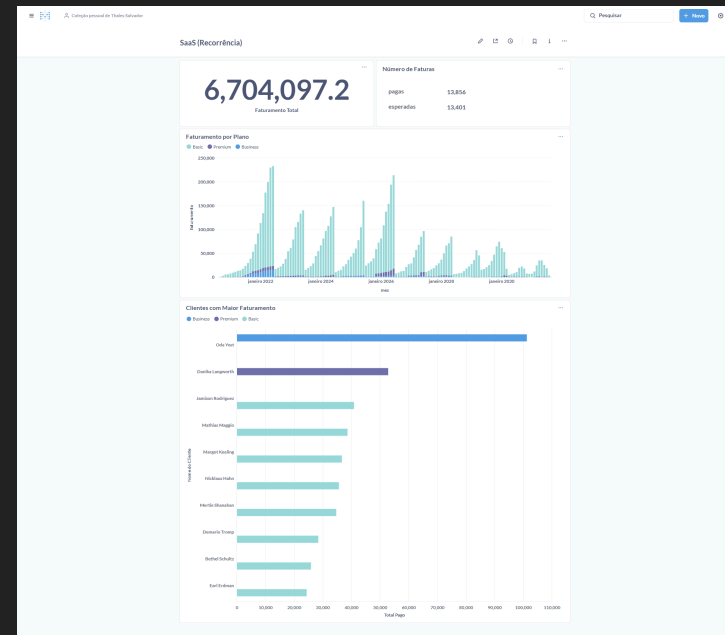
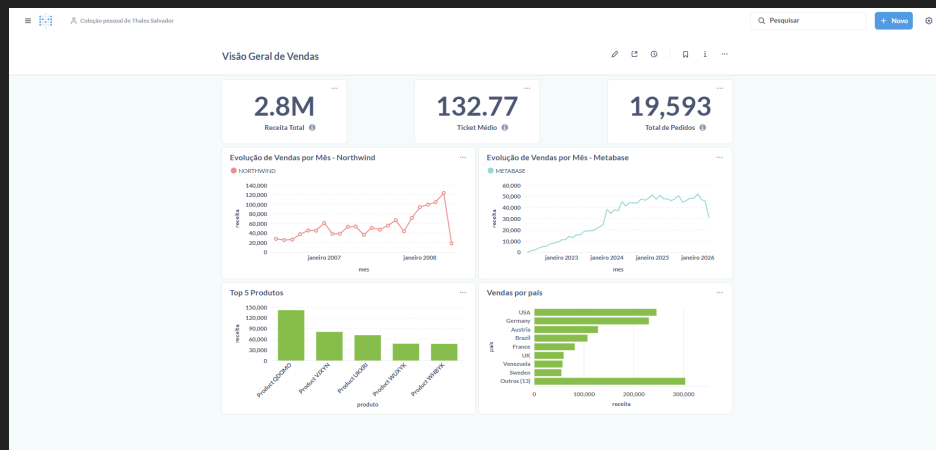
Data Marts

A camada final (Gold) entrega tabelas prontas para consumo (OBT - One Big Table):

- ▶ **dm_sales:** Visão geral de receita.
- ▶ **dm_b2b / dm_b2c:** Visões segmentadas por modelo de negócio.
- ▶ **dm_saas:** Métricas de recorrência e churn.
- ▶ **dm_product_analytics:** Performance de produtos vs avaliações.

dm_product_analytics	dm_saas	dm_b2b	dm_b2c	dm_sales
AZ record_type	AZ invoice_nk	AZ order_nk	AZ order_nk	AZ order_nk
AZ record_nk	AZ customer_nk	AZ source_system	AZ source_system	AZ source_system
AZ customer_nk	AZ plan_name	🕒 order_date	🕒 order_date	🕒 order_date
AZ product_nk	123 payment_amount	123 quantity	123 quantity	123 quantity
AZ metric_name	<input checked="" type="checkbox"/> expected_invoice_flag	123 gross_amount	123 gross_amount	123 gross_amount
123 metric_value	🕒 invoice_date	123 net_amount	123 net_amount	123 net_amount
🕒 metric_timestamp	AZ customer_name	AZ customer_name	AZ customer_name	AZ customer_name
AZ context_info	🕒 invoice_date	AZ customer_city	AZ customer_city	AZ customer_city
AZ customer_name	AZ customer_name	AZ segment	AZ segment	AZ segment
AZ segment	AZ segment	AZ customer_country	AZ customer_country	AZ customer_country
AZ product_name	AZ plan	AZ product_name	AZ product_name	AZ product_name
	123 year	AZ category_name	AZ category_name	AZ category_name
	123 month	AZ supplier_name	AZ supplier_name	AZ supplier_name
	AZ month_name	AZ employee_name	AZ employee_name	AZ employee_name
		AZ job_title	AZ job_title	AZ job_title
		AZ channel_name	AZ channel_name	AZ channel_name

Dashboards Metabase



Stack Tecnológico



Python 3.13

Ambiente virtual e gestão de dependências.



dbt-postgres 1.9

Modelagem, testes (Data Quality) e Snapshots.



PostgreSQL

Data Warehouse local com extensão FDW ativa.

** Utilizamos o PostgreSQL 12 pela facilidade de já termos instalado na máquina, mas o código é aderente a versões mais recentes.*

Dúvidas?

Documentação completa e código fonte:

github.com/thalessalvador/DW

Comandos principais:

```
dbt run --select tag:staging
dbt run --select intermediate
dbt snapshot --select products_snapshot suppliers_snapshot employees_snapshot customers_snapshot
dbt run --select dim_customer dim_product dim_supplier dim_employee dim_category dim_date dim_geography dim_channel fact_sales fact_invoices fact_feedback
fact_reviews fact_analytic_events dbt run --select dm_sales dm_b2b dm_b2c dm_saas dm_product_analytics
```

Comandos acessórios:

```
dbt run tests
dbt docs generate
dbt docs serve
```

Alunos

Thales Augusto Salvador

Carlos Henrique Barbosa da Silva