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

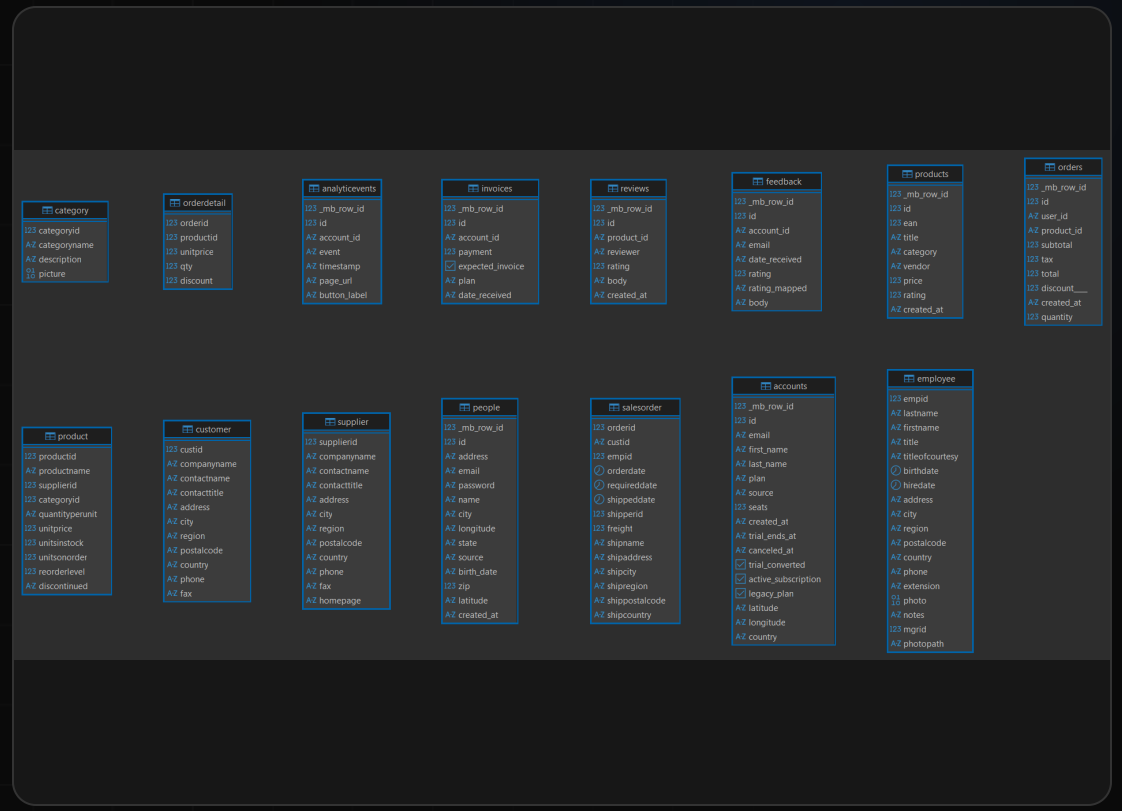
Unificar essas origens díspares em um único **Data Warehouse** no PostgreSQL, utilizando **dbt** para orquestração e transformação, habilitando análises cruzadas de produtos e performance de vendas.

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.



Pipeline de Transformação (DAG)

Raw (FDW) → Staging (Padronização) → Intermediate (Consolidação) → Snapshots (SCD2) → Fatos/Dimensões.

Gráfico interativo gerado pelo comando `dbt docs generate` seguido pelo comando `dbt docs serve`

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'

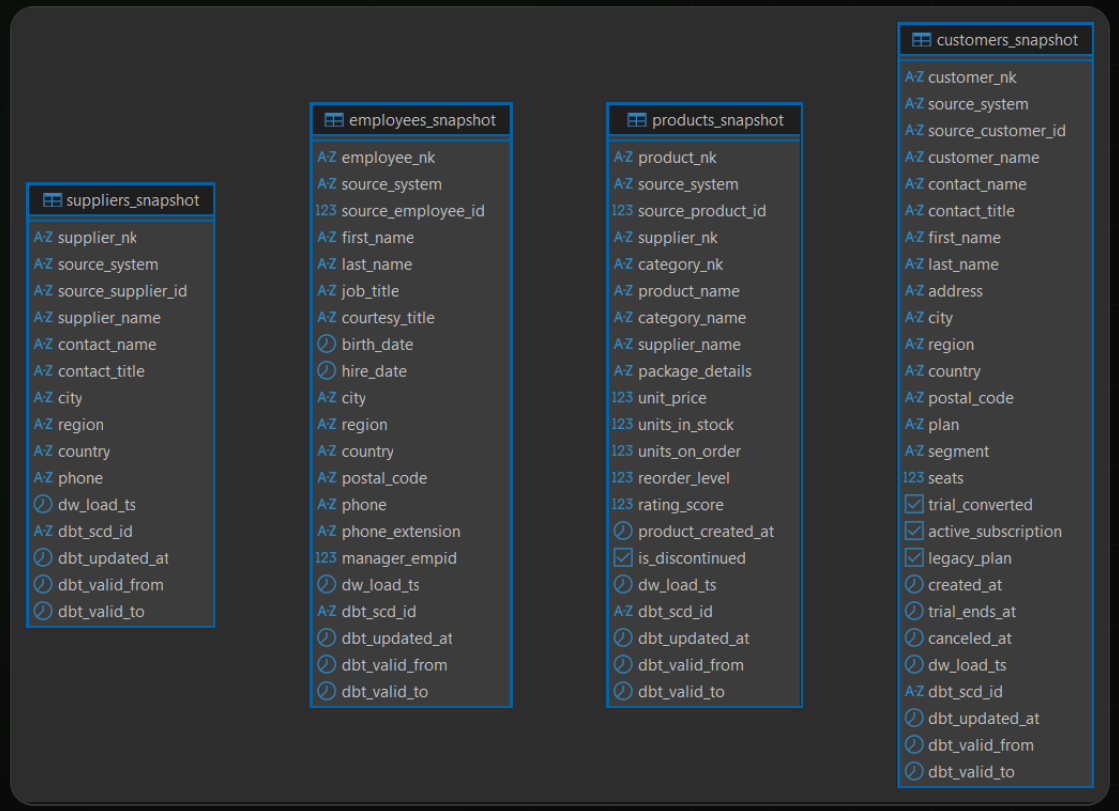
Adotamos a estratégia de snapshot `check` do dbt.

- 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



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.

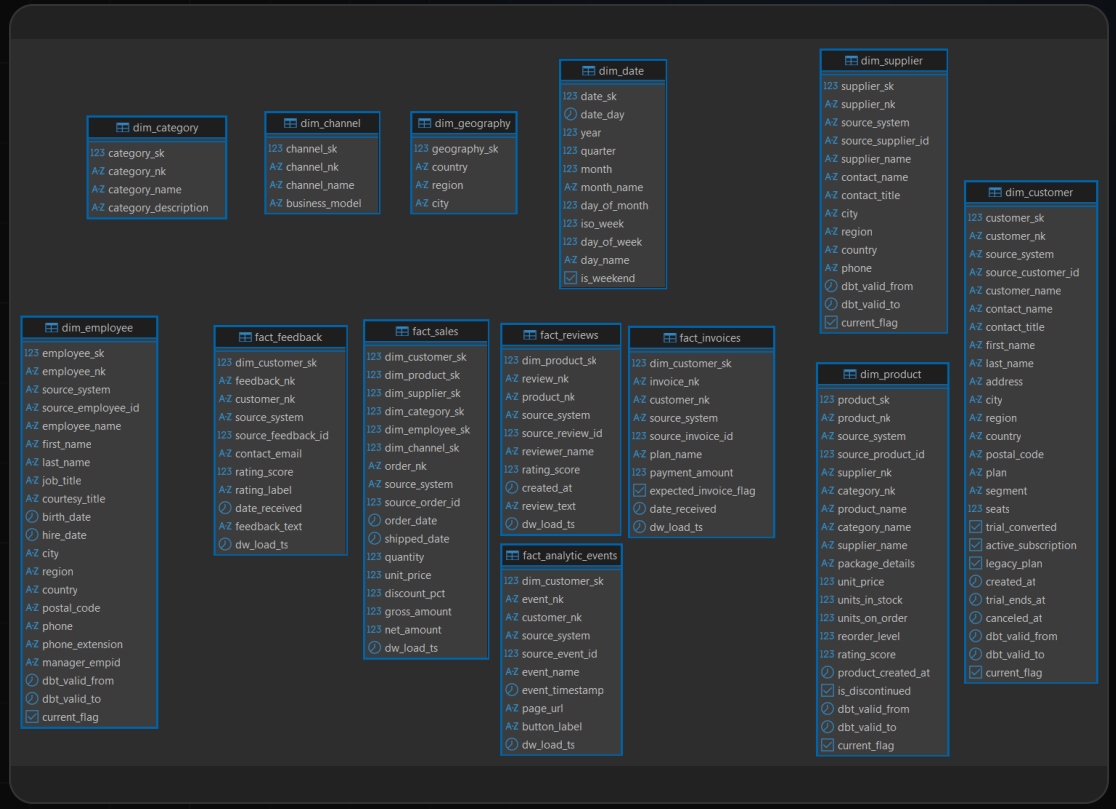
Modelagem Star Schema (DW Core)

Dimensões Unificadas

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

Fatos de Negócio

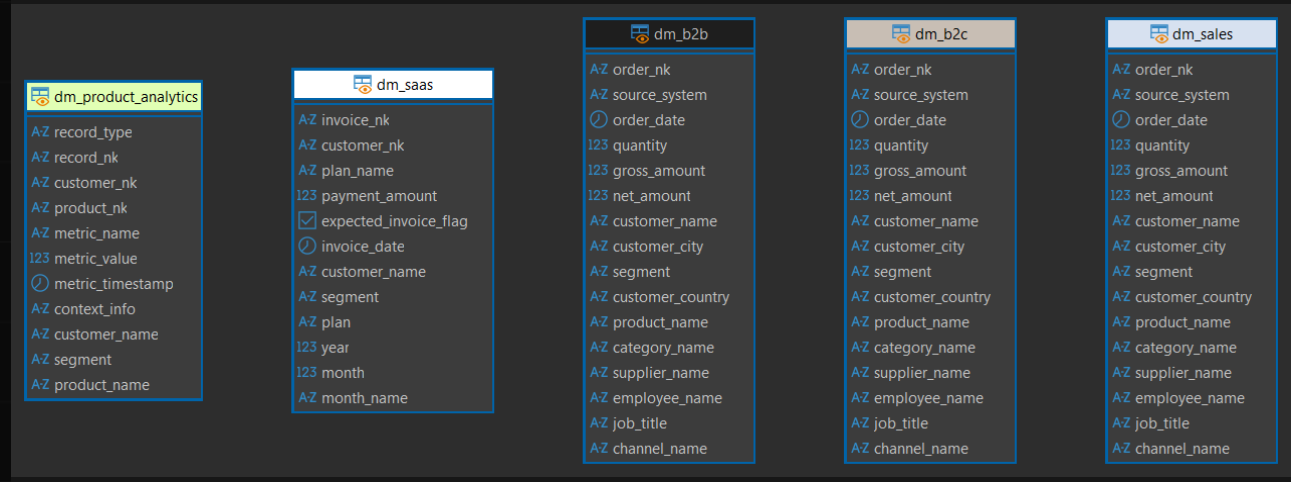
- `fact_sales`: Vendas consolidadas.
- `fact_analytic_events`: Eventos de clique/navegação do SaaS.



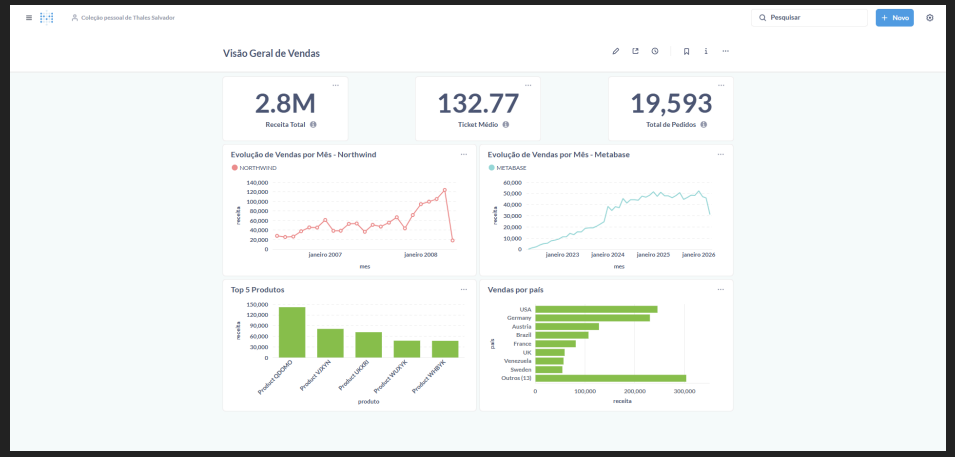
Data Marts

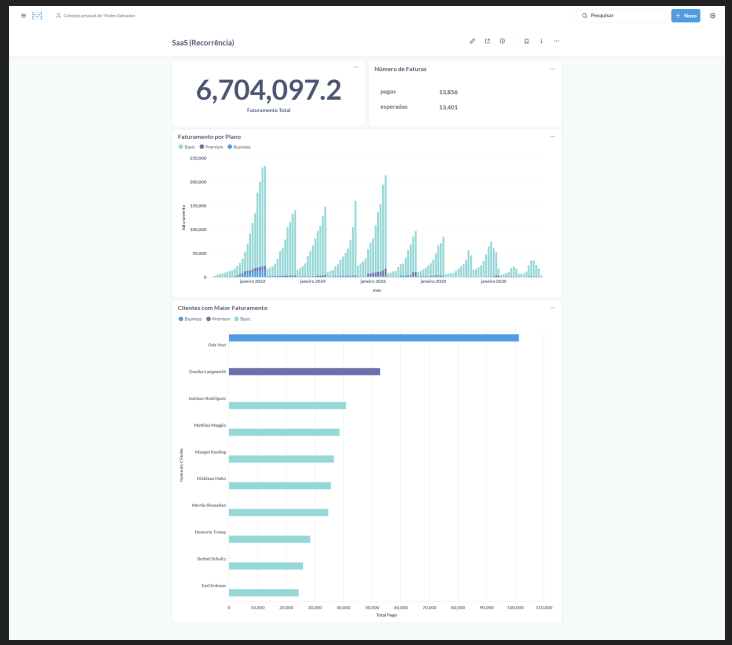
A camada final (Gold) entrega tabelas prontas para consumo (OBT - One Big Table):

- **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.



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