

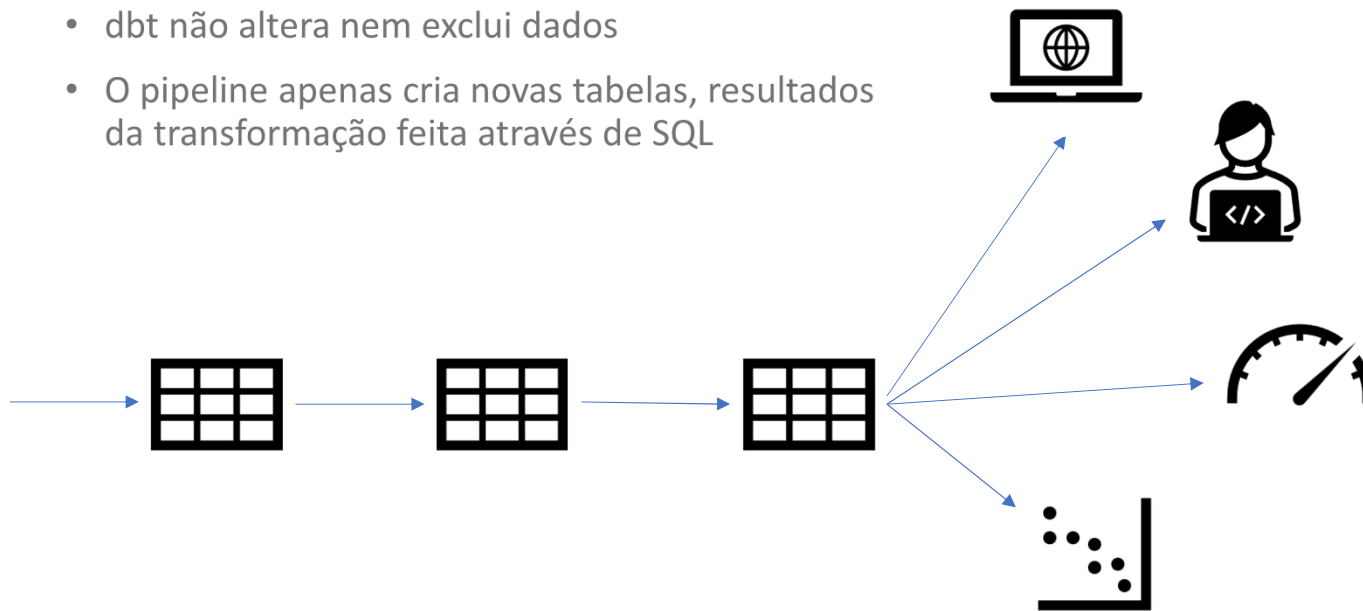
dbt

- Baseado em Modelos que formam um Pipeline
- Um modelo é um arquivo SQL, com extensão SQL, que é executado
- Um modelo gera uma única tabela como saída
- Um modelo pode ter outros modelos como entrada, e servir de entrada para outros modelos

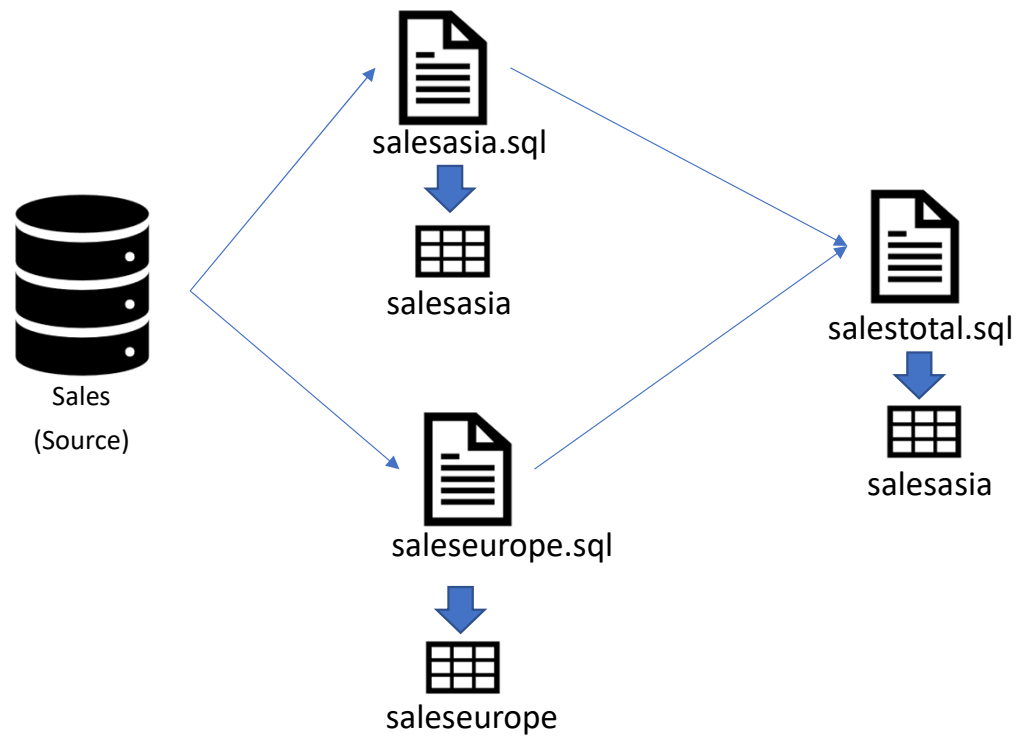


dbt

- dbt não altera nem exclui dados
- O pipeline apenas cria novas tabelas, resultados da transformação feita através de SQL

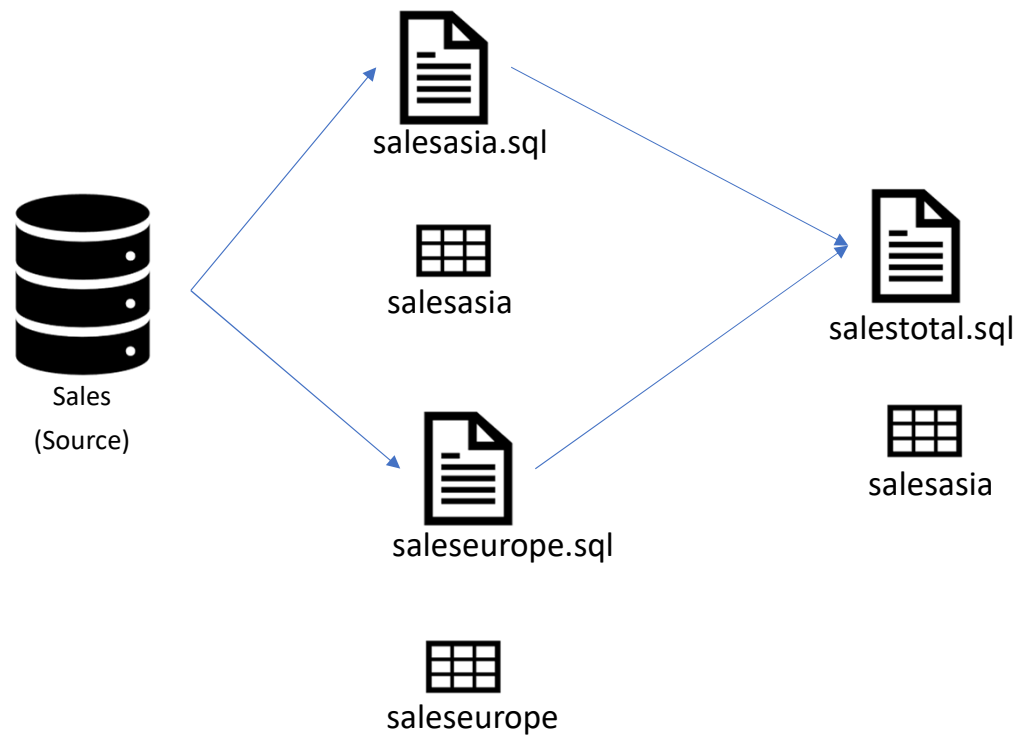


Exemplo

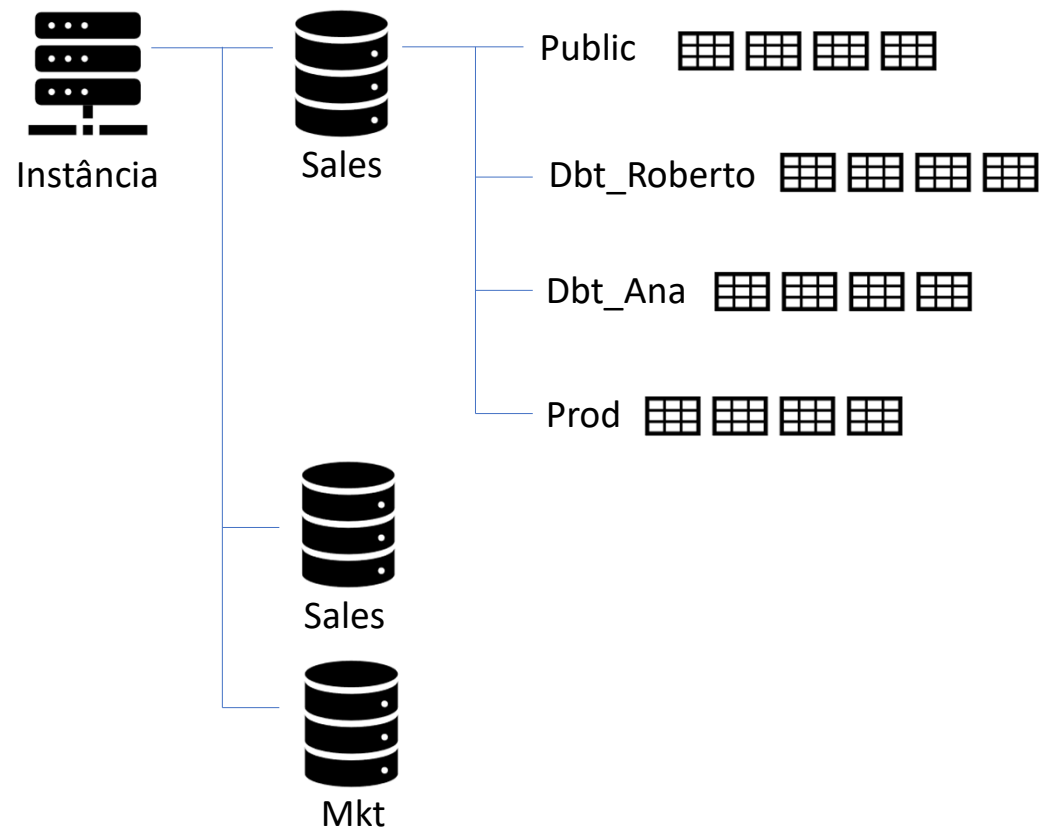


Materializations

- Table
- View
- Incremental
- Ephemeral



Schemas: Desenvolvimento, e Deploy



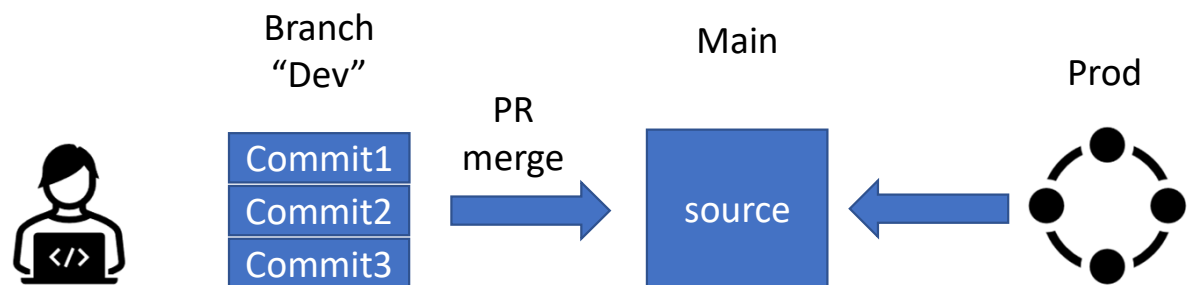
Run

- Desenvolvimento
 - Você pode executar todo o pipeline
 - Você pode executar modelos de forma individual
- Deploy
 - O modelo é executado através de um Job
 - O Job pode ser agendado ou executado manualmente



Projetos

- Dbt é baseado em projetos, como qualquer software
- É versionado em uma ferramenta como Github
- Um projeto tem um repositório
- Um desenvolvedor faz commits para uma branch
- PRs fazem merge no branch master
- Quando você roda o projeto em produção, ele pega a versão da branch master



SQL e Jinja2

- Dbt executa SQL
 - Qualquer SQL do banco de dados é válido
- Jinja pode ser utilizado
 - Linguagem de templates
 - De forma simplificada: variáveis que são substituídas na execução
 - Substitui o template em tempo de execução
 - No final, tudo é SQL!

Source

- São as fonte dos dados que serão transformadas
- Redshift, Snowflake, Postgres etc.