Dia 04

Revisão - Dia 03

DBT - Tags

DBT - Tags

- **Definição:** Etiquetas que você pode aplicar a modelos, testes, snapshots e análises no dbt.
- Usadas para organizar e agrupar objetos relacionados.

DBT - Tags

- Facilita a execução de um subconjunto de modelos.
- Ajuda na organização e documentação.
- Permite a configuração específica para grupos de modelos.

Demo

...

DBT Tags

DBT Select Syntax

...

https://docs.getdbt.com/reference/node-selection/syntax

Dúvidas

DBT Cloud

Primeiros passos com DBT

https://www.youtube.com/watch?v=ZPYmgt4zE7M

Demo

...

DBT Cloud

Dúvidas

DBT - Seed

DBT - Seeds

- São arquivos csv na pasta Seed (Seed/*.csv)
- Serve para construir tabelas com pouca quantidade de dados, geralmente tabelas de referência/arquivos estáticos, que não mudam com frequência (SCD - Tipo 0).
- Não são recomendadas para tabelas grandes e que atualizam com frequência.
- Não são recomendadas como recurso de carga de dados para dentro do data warehouse (EL).

DBT - Seeds

- Seeds são materializadas com o comando `dbt seed`, como isso o DBT materializa os arquivos CSVs como tabelas no data warehouse.
- Seeds podem ser referenciadas com o comando `{{ ref('<seed>') }}`
- É possível documentar Seeds com um arquivo *.yml dentro da pasta Seed (Seed/*.yml).
- Exemplos:
  - Dados geográficos (estado, cidade, etc.)
  - Dados de tempo (Feriados, dias, etc.)

Dúvidas

DBT - Analyses

DBT - Analyses

- São arquivos sql dentro da pasta Analyses (Analyses/*.sql).
- Analyses são consultas que não serão materializadas (`dbt run`), diferente dos modelos. Mas podem ser compiladas utilizando `dbt compile`.
- Utilizada como caixa de areia (sandbox).
- Úteis para treinamentos, auditorias, testes manuais, e consultas avulsas. Muito utilizada para testar modelos antes de criá-los
- Suportam utilização de Jinja e comandos DBT (referencias, macros, templates, etc.).

DBT - Analyses

- Criação de testes, modelos
- Análise de dados/resultado
- Criação de consultas para dashboards
- Armazenamento de scripts/Store procedures (e.g. criação de usuário, consultas de sistema, etc.)
- Consultas de treinamento (analyses não afetam os modelos e não são materializadas)
- Auditoria e refatoração (e.g. testar se uma query performa melhor que a outra, verifica se o novo modelo e o legado retornam o mesmo resultado,etc).

DBT - Primeiros comandos

- dbt <comando> --threads=<n_threads>
- dbt compile
- dbt run
- dbt seed

Dúvidas

Jinja

Jinja - Templates

- São documentos de texto onde uma parte ou o todo do conteúdo é gerado automaticamente.
- Aceita qualquer arquivo de texto (html, xml, text, sql, etc.)
- Os dados são armazenados separadamente e automaticamente adicionados ao documento.
- Herança de templates permite a utilização de templates dentro de templates, seja por meio de inclusão ou por meio de criação de templates de base.

Jinja

- Biblioteca python de templating (e.g. email de marketing).
- Auxilia na implementação do princípio DRY (Don’t Repeat Yourself).
- Traz aspectos de linguagem funcional e princípios de engenharia de software para arquivos de texto (e.g. SQL, YAML, MD)
- Permite a escrita do mesmo código, porém com comportamentos diferentes a depender do ambiente (e.g. desenvolvimento, produção).
- Auxilia na diminuição da quantidade de código escrito (mais conciso).

Jinja

- Permite a criação de funções (Macros)
- Auxilia o DBT no entendimento das propriedades e relacionamentos entre as tabelas, além de empoderar a linguagem SQL.
- Permite a utilização de estruturas de dados de python (e.g. tuplas, listas, dicionários, etc.)

Jinja - Benefícios

- Acelera o desenvolvimento de código SQL
- Melhora a colaboração entre a equipe
- Permite comportamento específico para cada ambiente
- Auxilia no controle de permissões ao data warehouse
- Auxilia na remoção e obsolescência de tabelas no data warehouse
- Permite a escrita de códigos SQL mais concisos

Jinja - Delimitadores

- {{ expressões }}
- {% declarações %}
- {# comentários #}

Jinja - Expressões

- {{ variável }}
- {{ variável | tipo_dado }}
- {{ variável | filtro }}
- {{ string ~ string ~.. }} - concatenação de string

Jinja - Filtros

Maths: abs, int, float, round, sum

Str: capitalize, length, center, escape, lower, regex

Lists: join, last, sort, shuffle, json_query

Jinja - Declarações

- {% set <variável> = <valor>/<estrutura de dados> %}
- {% set <variáveis> % }..{% endset %}
- {% if %}..{% endif %}
- {% if %}..{% else%}..{% endif %}
- {% for %}..{% endfor %}
- {%do <operacao>%}

Jinja - Estrutura de decisão

```
{%if opcao == 1%}
    Você selecionou a opção 1
{%elif opcao == 2%}
    Você selecionou a opção 2
{%else%}
    Você não selecionou nenhuma opção
{%endif%}
```

Jinja - Estrutura de repetição

```
{%for item in lista%}
    {{item}}
{%else%}
    Nenhum item encontrado
{%endfor%}
```

Jinja - Extras

{# <comentario> #}

{%- .. %} - limpa todos os espaços antes

{% .. -%} - limpa todos os espaços depois

{%- .. -%} - limpa todos os espaços antes e depois

Jinja - Funções

- range
- loop (last, first, cycle)

DBT - Jinja

DBT - Jinja - Variáveis

- {{ this }} - variável contendo informações dos objetos mapeados no modelo atual.
- {{ target }} - variável contendo informações do data warehouse (e.g. connection, warehouse, profile_name, name, schema, type, and threads)
- {{ execute }} - variável dbt que é verdadeira se o SQL de um modelo está sendo executado durante o runtime (dbt run) e não apenas sendo compilado (preview).

DBT - Jinja - Funções

- run_query(‘<consulta>’): executa uma consulta e salva os resultados
- log(‘<texto>’, default): adiciona uma linha de texto nos logs do dbt, utilize o parâmetro default=True para adicionar à CLI também.

DBT - Jinja - Objetos

- Table
- Relation

DBT - Jinja - Dicas

- Escreva em SQL puro
- Refatore com Jinja
- Remova espaços em branco

Demo

...

DBT Jinja

Dúvidas

DBT - Macros

DBT - Macros

“Considere o caso em que temos três modelos que usam a mesma lógica. Poderíamos copiar e colar a lógica entre esses três modelos. Se quisermos mudar essa lógica, precisamos fazer a mudança em três lugares diferentes. As macros nos permitem alterar a lógica em um só lugar e atualizar automaticamente todos os três modelos.”

DBT - Macros

- São funções escritas em Jinja, salvas em arquivos sql na pasta Macro (Macro/*.sql)
- Permite escrever códigos genéricos e parametrizáveis, que podem ser referenciados no projeto.
- Para testar as macros é possível utilizar o comando `dbt run-operation <macro>`
- Atentar-se para a ponderação entre legibilidade e reusabilidade
- Auxilia na colaboração e compartilhamento de código

DBT - Macros - Exemplos

- Transformações comuns de colunas (e.g. datas, texto, números, etc.)
- Declarações de SQL comuns (e.g. limit, distinct, order by, etc.)
- Declarações específicas de ambiente (e.g. puxar apenas dados do último dia para ambiente de desenvolvimento)
- Conceder acesso para tabelas e views em determinado esquema do data warehouse
- Personalizar o comportamento do DBT baseado no ambiente (e.g. dev, prod)
- Limitar o número de linhas automaticamente para economizar computação
- Usar diferentes databases ou esquemas para as Sources
- Alterar a severidade de um teste de acordo com o ambiente (e.g. warn, error)

DBT - Macros

- Crie um código SQL
- Adicione o código a uma macro sem argumentos
- Parametrize a macro com argumentos
- Defina valores padrões para os argumentos (quando necessário)
- Remova espaços em branco

Demo

Macros

DBT - Packages

DBT - Packages

- Permitem a importação de modelos, seeds, analyses, e macros de outros desenvolvedores
- São fontes definidas em uma arquivo packages.yml
- É possível importar pacotes utilizando o comando `dbt deps`
- O comando acima vai até a fonte especificada, valida o código, clona, e instala todas as dependências necessárias
- Auxilia o trabalho assíncrono e o compartilhamento de código
- Particularmente bom para dados padrões de ferramentas de EL (e.g. snowplow, airbyte, etc.)
- Particularmente bom quando há mais de um projeto no mesmo repositório

DBT - Packages

- É possível encontrar pacotes em hub.getdbt.com
- Geralmente códigos oficiais são mais performáticos
- É possível rodar comandos apenas para pacotes específicos utilizando `dbt <comando> –select package:<nome_pacote>`

DBT - Packages - Populares

- dbt_utils: diversas funções úteis para projetos dbt

Demo - Packages

...

hub.getdbt.com

DBT - Refatoração

Demo

Projeto pt.1

Dúvidas