Contents

Noteson technical demonstration of ingest nodes

These notes should be read in conjunction with the session recording.

Agenda

- Review of last session and context setting for this one
  - Previous session was focussed on the "Why", today we will focus on the "How".
  - We will review main tenants of the "Why" before moving on.
- Demo of standard DBT project: DBT-based ingest node "Dummy Jaffle".
- Demo of DBT project with automate_dv DBT macro extension: "Dummy Bridgetown".

We will demonstrate aspects of:

- DBT-based ingest node
  - best suited for OLAP use case
  - can ingest data through `dbt seed`
  - can ingest data from cloud storage through `COPY INTO` statement
  - can model data using DBT models
  - can use macro packages such as automate_dv to simplify data vault building
- (did not get to this) DBMate-based ingest node
  - best suited for OLTP use case
  - can perform database schema migrations through the use of dbmate

Definitions

- Cookiecutter - a template system for configuring projects based on questions asked and answers given.
- Jinja2 - template language used by Cookiecutter.
- Docker - container system used to bundle code and execute locally and remotely.
- Databricks Serverless SQL Warehouse - serverless cloud storage substrate and query engine supported by the DBT-based ingest node.
- PostgreSQL - RDBMS supported by the DBT and DBMate ingest nodes.
- DBT - data build tool based on Jinja2 responsible for the main work that the DBT-based ingest node performs.

- DBMate - light-weight “database schema migration” tool used in the DBMate-based ingest node.
- Database Schema Migrations - industry standard way of cumulatively applying changes to a database schema as part of a deployment pipeline.
- Bash - scripting system on unix/linux.
- ADO Pipelines - YML-driven CI / CD system on Azure.
- Azure Container Registry - a registry system for housing docker images on Azure.
- Databricks Personal Access Token (PAT) - a UUID-like token that you can configure on Databricks on a per-workspace basis in order to gain access to resources.
- DBT-based ingest node - a system with DBT at it’s heart for developing data ingest and transformation logic to execute locally or remotely, and to target a storage substrate such as DBR or PGSQL particulary well suited for ELT-type work for OLAP use cases.
- DBMate-based ingest node - a system with DBMate at it’s heart for developing database schema migrations to execute locally or remotely, and to target a RDBMS storage substrate such as MySQL or PGSQL particularly suited for supporting a OLTP system.
- Data Vault - a write-optimised, long-term storage, changed tracked normalised data modelling methadology to fill the gap between raw data and data warehouses.
- automate_dv - DBT macro package foor simplifying data vault building.
- Data Warehouse - a read-optimised, domain-specific / domain-analytics-specific data modelling methadology for supporting OLAP workloads.
- Star Schema - a popular data modelling technique for creating a data warehouse.

How things relate

This is an attempt at relating things in this space.

Figure 1: diagram of how things relate

Cookiecutter run

- All of the answers below to the Cookiecutter questions are obtained from ADO and Databricks.

- The ADO config file is something you have to setup in your home directory.
- The ADO config file is only required for the ADO Pipeline automation, at the time when you want to setup the build, release and schedule pipelines.

```
cookiecutter git@ssh.dev.azure.com:v3/exploreai/CORE.Utilities/CORE.Meshnodes.BaseTemplates/
ingest_node_name [Ingest Node Template]: dummy_jaffle
az_devops_repo [https://dev.azure.com/exploreai/CORE.Utilities/_git/CORE-POC1]: https://dev.
Select data_substrate:
1 - databricks
2 - postgres
Choose from 1, 2 [1]: 1
databricks_workspace [https://adb-891777510264692.12.azuredatabricks.net/]: https://adb-4472
database_host [adb-891777510264692.12.azuredatabricks.net]: adb-4472170994427587.7.azuredata
databricks_sql_warehouse_path [/sql/1.0/endpoints/a077556ed384ed67]: /sql/1.0/warehouses/22a
databricks_catalog [null]: dummy_jaffle
database_user [psqladmin]:
database_port [5432]: 443
database_threads [4]: 5
profile_name [data_vault]:
azure_storage_container_url [wasbs://adf-pipeline-demo@datavalidation.blob.core.windows.net/
ado_pipelines_build_agent_pool_name [CORE Pipelines]: Default
ado_pipelines_folder_name [CORE]: dummy_jaffle
ado_profile_name [CORE]: DEFUALT
scheduled_release_cron [0 6 * * *]:
----------
Template clone successful
----------

--> Attempting to move pipeline definitions to relevant location...
        --> Moving files...
        --> File move successful.
--> Searching for ADO config file in home dir...
? ADO config file detected. Would you like to alter/extend it? No
User declined to modify ADO config file at /home/kerneels/.adocfg. Skipping file generation

--> Generating `.env` file from supplied definition file...
        --> Generation successful.
        --> Please inspect/edit the generated file at `dummy_jaffle/.env` to ensure that th
```

These notes should be read in conjunction with the session recording.