Pipeline orquestration*

Distributed Data Processing Environments

Lab Guide 7

This session aims at creating data processing pipelines with SQLMesh.

Steps

1. Install SQLMesh:
   - Alternative 1: Create and activate a Python virtual environment. Execute: `pip install sqlmesh[duckdb]`
   - Alternative 2: Deploy attached Vagrantfile to Google Cloud and connect with SSH.
2. Download the Bundesliga dataset from HuggingFace.[1]
3. Create a SQLMesh project with: `sqlmesh init`
4. Create a new external model that reads the dataset and import it with: `sqlmesh create_external_models`
5. Update data with: `sqlmesh plan`
6. Create an audit that verifies that for every winning match, the number of goals for is greater that goals against.
7. Create an new model that computes a summary of goals for and agains, number of wins, losses, and draws for each team in each season.
8. Create a new model that computes the top team according to goal average for each season.
9. Observe the pipepline with: `sqlmesh ui`
10. If using Alterntive 2, remember to destroy the VM and cleanup with: `vagrant destroy`

Learning Outcomes Recognize situations where data processing can be expressed as a pipeline. Applica orquestration tools to automate complex data processing operations.

*May use of AI tools for steps 6 to 8.

[1] `https://huggingface.co/datasets/KrisSommer/Bundesliga_Stats_2018-2024`

Cheat Sheet

```
MODEL (
  name somename,
  kind FULL,
  cron '@daily',
  grain somecol,
  audits (someaudit),
);

SELECT ... FROM ...
```