TD7: Ingeniería de Datos

Trabajo Práctico Grupal

Segunda entrega: arquitectura y flujos de datos

Esta entrega está enfocada en ejercitar los conceptos revisados en la segunda parte de la materia. El trabajo consiste en tomar el modelo implementado en la primera entrega y enmarcarlo en el contexto de una organización que tiene esa base de datos, por lo cual será necesario diseñar una arquitectura, flujos de datos y procesos de gobernanza.

Se espera que el entregable principal sea un **informe** que detalle cómo es la resolución de cada punto, incluyendo la descripción de los flujos implementados, el código y los casos de uso relevados. Además del informe, se esperan los archivos de soporte (código python, dags, transformaciones en dbt, etc.), preferiblemente un repositorio **privado** o, como segunda opción, un zip.

1. **Arquitectura**. Diseñar una arquitectura que encuadre el dominio en el contexto de una organización.
   Además de los datos pensados en la primera parte, enunciar si se consideran datos de otras fuentes adicionales.
   Contemplar:
   a. ¿Cuál es el origen de los datos? ¿Qué cadencia tienen, con qué formato arriban, qué volumen se espera?
   b. ¿Cómo es el linaje de los datos desde su origen hasta que las distintas aplicaciones los consumen? ¿Cómo son los procesos intermedios y qué patrones siguen?
   c. ¿Cuáles son los usos que tienen los datos en las distintas etapas de procesamiento?
   d. ¿Cómo es la gobernanza del dato? ¿Cuáles son los roles y qué tipos de permisos tienen sobre los datos?
2. **Flujo de carga de datos**: implementar un pipeline de carga de datos para las tablas armadas en Postgres en la primera entrega. Se espera tener un DAG armado en Airflow de cadencia horaria o diaria (según la justificación) con la carga de datos.
   a. Opción sintética: se puede utilizar Faker para generar datos sintéticos en la inserción.
   b. Opción dataset: se pueden aprovechar conjuntos de datos que existan en repositorios públicos; sin embargo, se espera que se pueblen las tablas progresivamente y que el job no sea cargar

el dataset entero. Se puede combinar con Faker para las entidades de las cuales no se encuentren datos.

c. Observaciones sobre el DAG: el DAG debe incluir al menos cuatro nodos y no puede ser una lista enlazada.

3. **Enriquecimiento**: armar al menos dos transformaciones con DBT y orquestar su ejecución con Airflow.
   Las transformaciones **deben tener un caso de uso** justificado en el informe. El tipo de caso de uso puede ser analítico, reporte o utilizado por una aplicación particular.
   Se debe elegir un tipo de materialización y debe ser justificado en el informe.
   Se deben configurar los sources aparte de los modelos en el YAML (ver ejemplo).
   Donde tenga sentido, usar también los tests out-of-the-box que ofrece DBT (ejemplo).
   Bonus (opcional, suma 1pto): agregar un test personalizado para los datos (guía) con un caso justificado en el informe.

Repositorio

Como base para el TP les armamos un repositorio base que ya tiene configurado Airflow, DBT, Postgres, tiene las conexiones ya realizadas y un setup general del ambiente. El proyecto ya tiene un ejemplo de cómo utilizar Faker para cargar datos de un dominio de ejemplo, al igual que una modularización inicial -- pero siéntanse libres de modificar lo que necesiten del código.